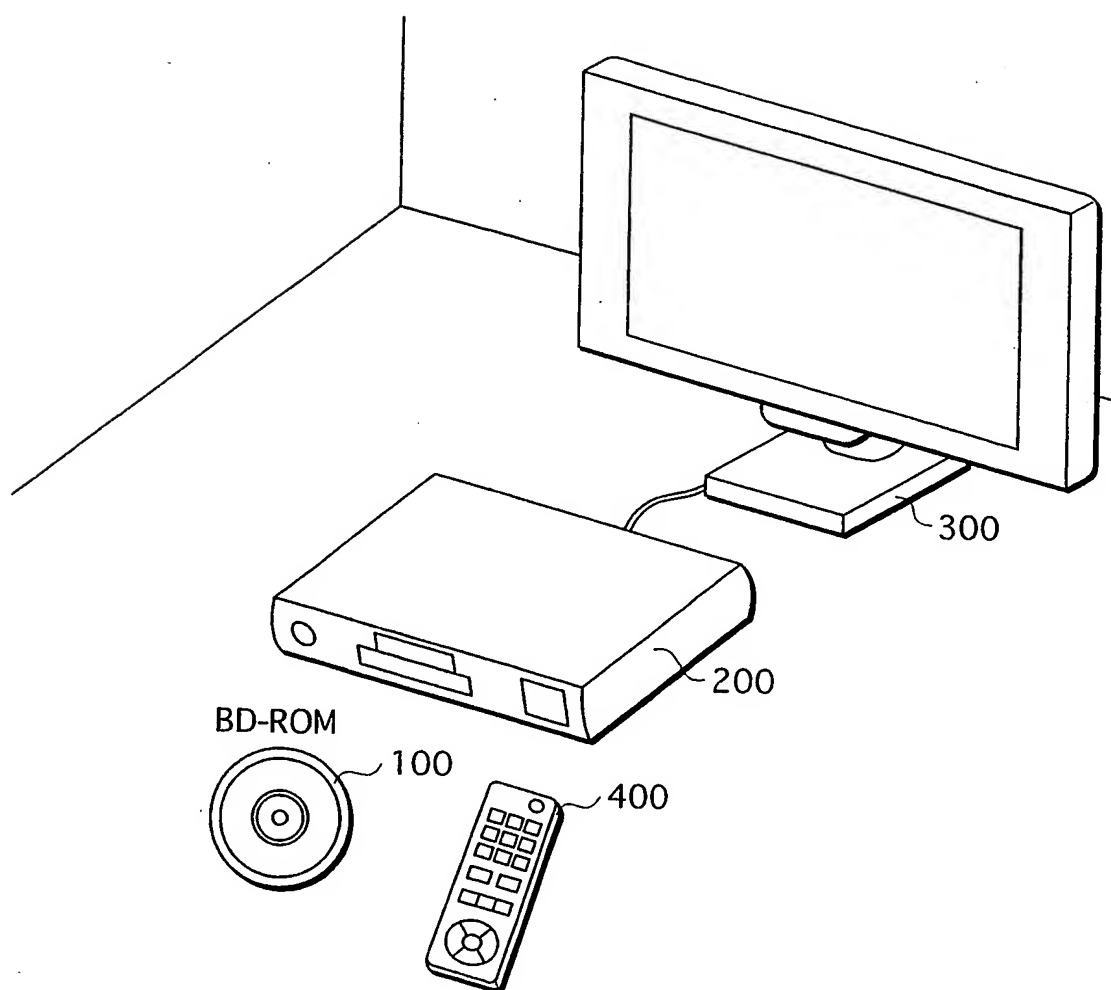
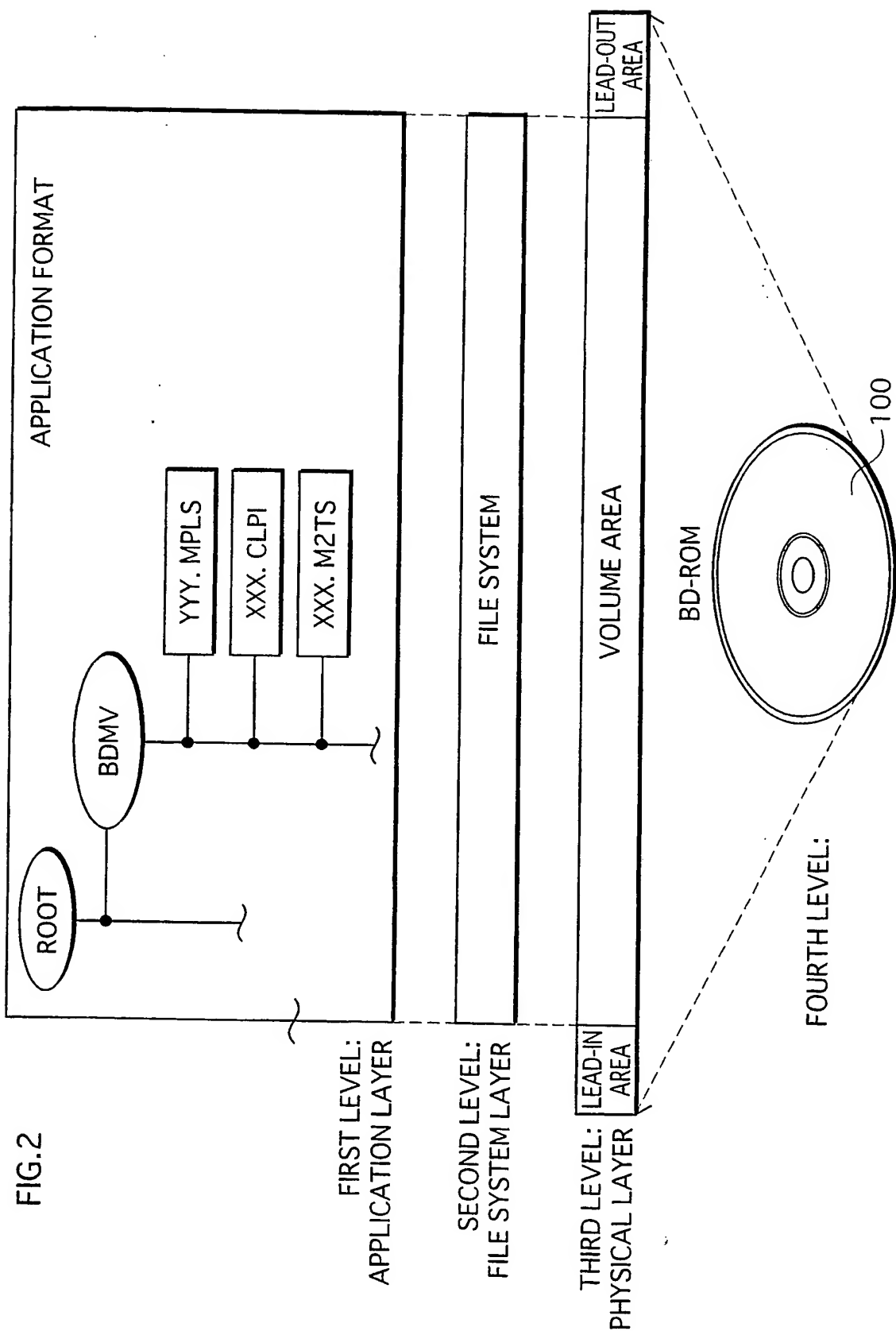


FIG.1





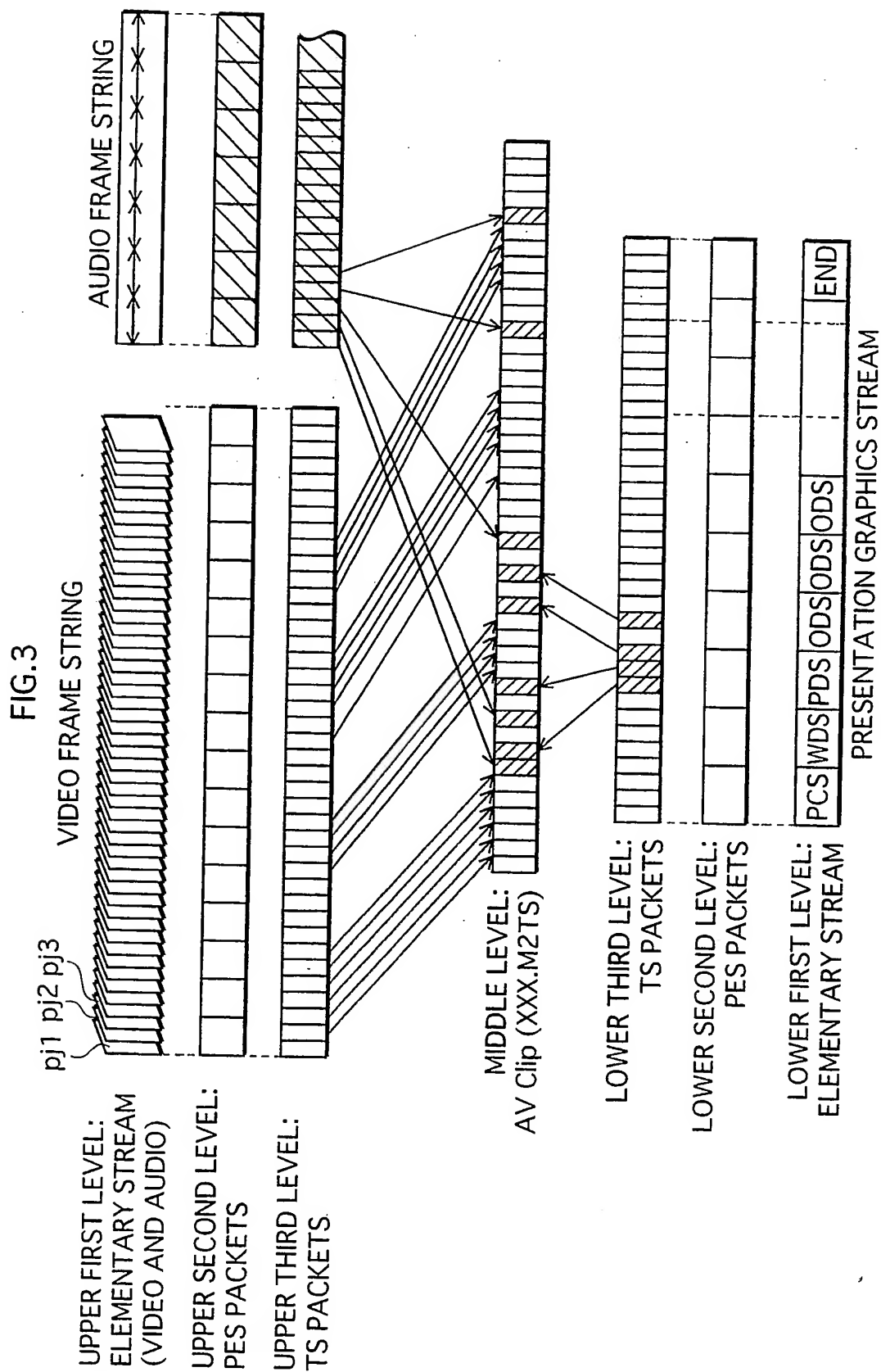


FIG. 4A

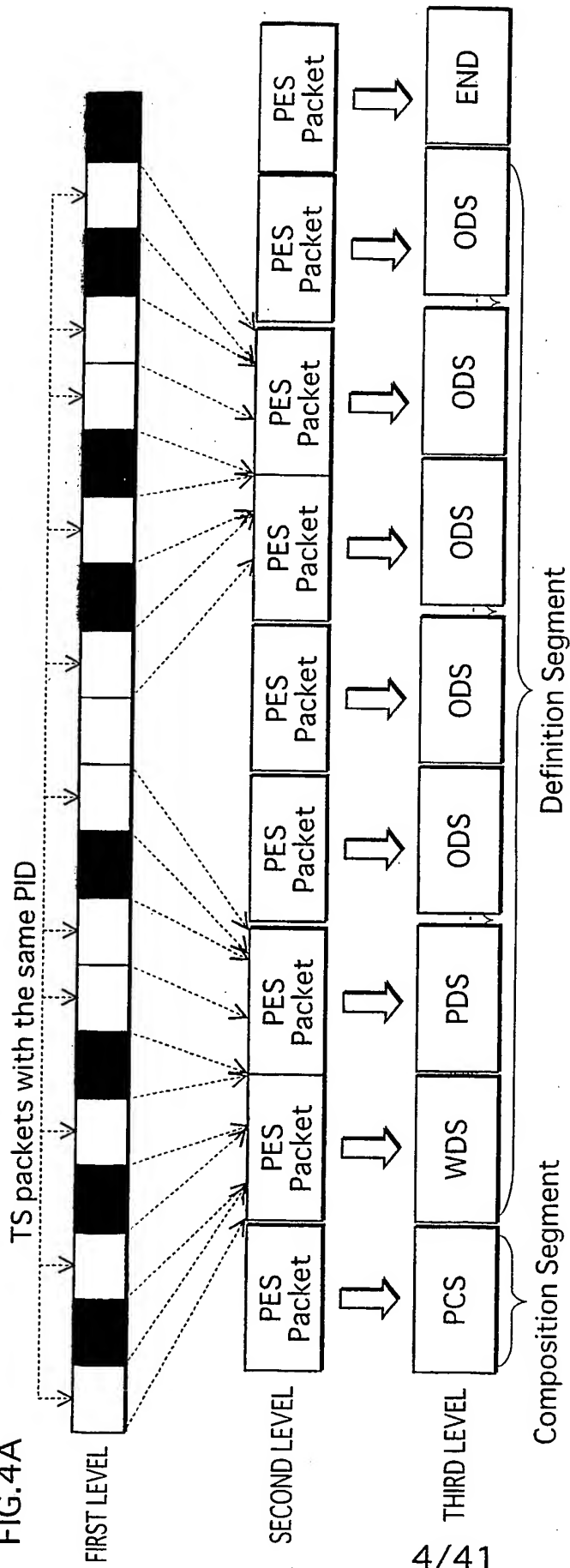


FIG. 4B

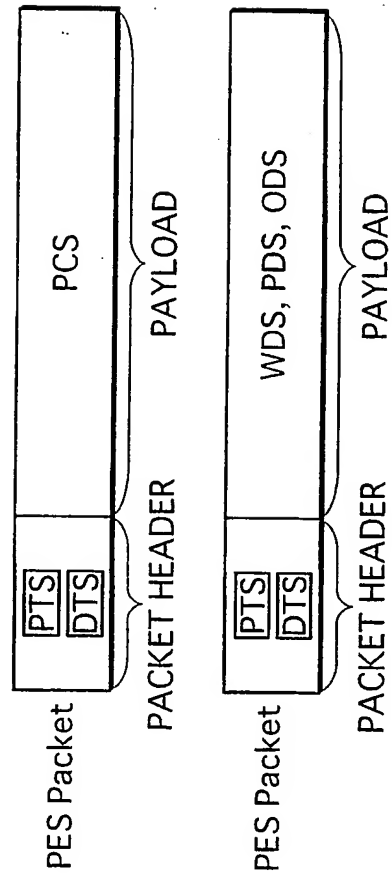


FIG.5

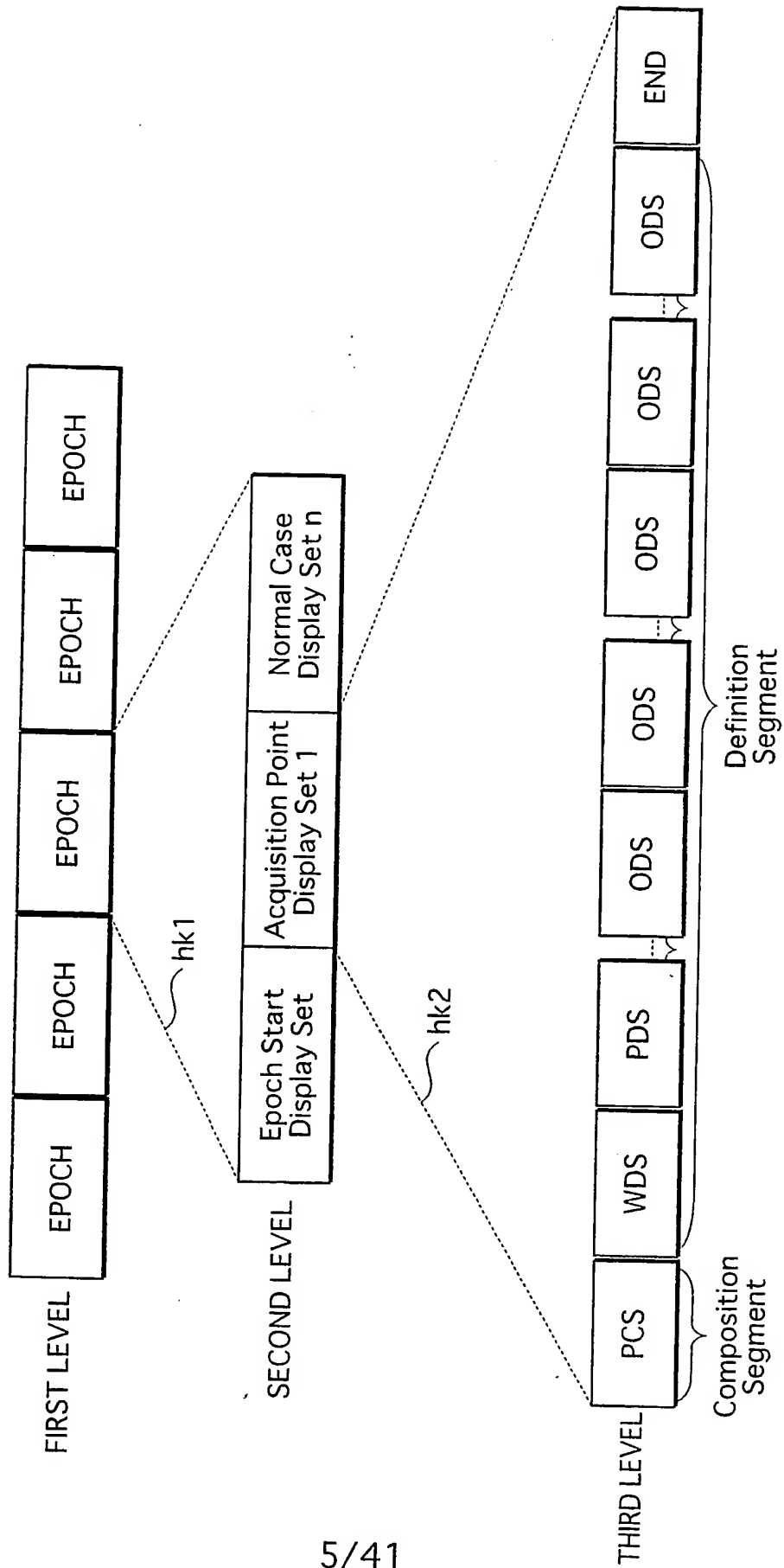


FIG.6

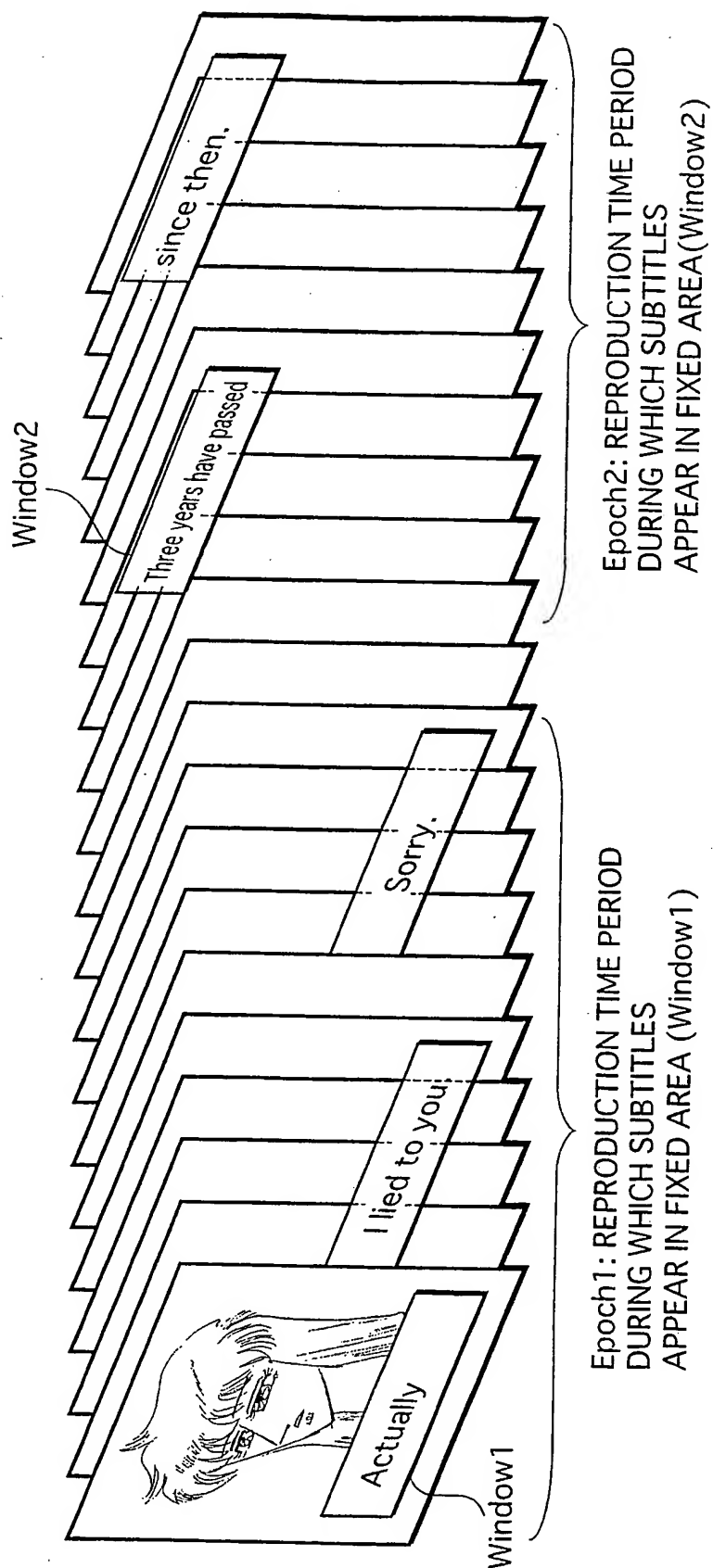


FIG.7A

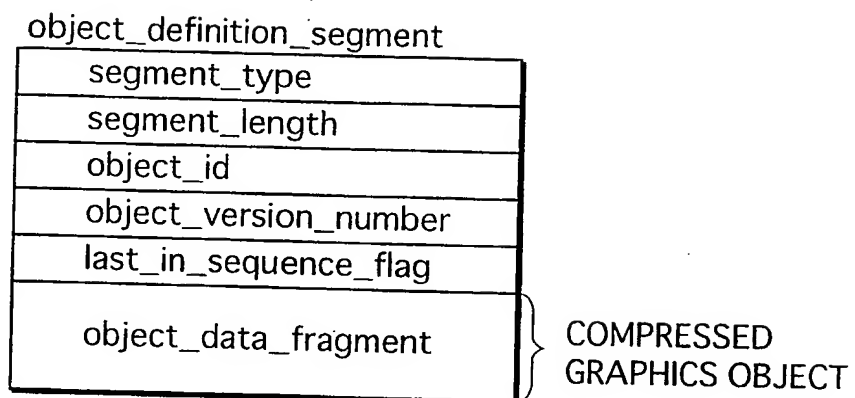


FIG.7B

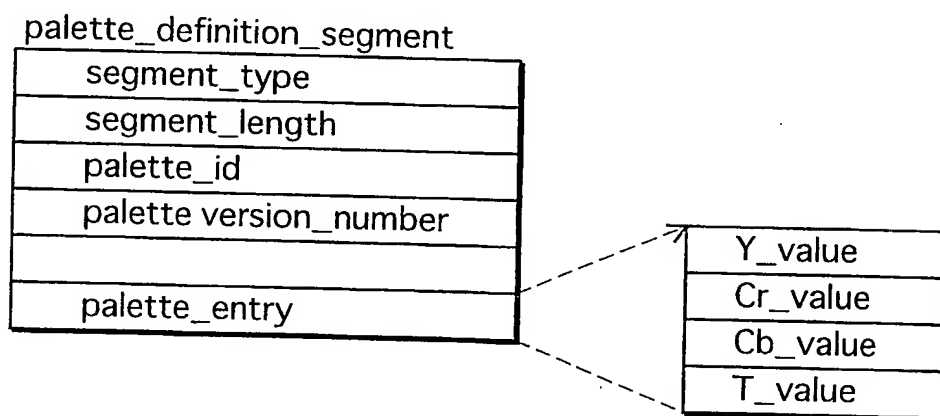


FIG.8A

window_definition_segment

window_id
window_horizontal_position
window_vertical_position
window_width
window_height

FIG.8B

presentation_composition_segment

segment_type
segment_length
composition_number
composition_state
palette_update_flag
palette_id
composition_object(1)
composition_object(2)
:
composition_object(i)
:
composition_object(m)

wd1

object_id
window_id
object_cropped_flag
object_horizontal_position
object_vertical_position
cropping_rectangle INFORMATION (1)
cropping_rectangle INFORMATION (2)
:
cropping_rectangle INFORMATION (i)
:
cropping_rectangle INFORMATION (n)

wd2

object_cropping_horizontal_position
object_cropping_vertical_position
object_cropping_width
object_cropping_height

FIG.9

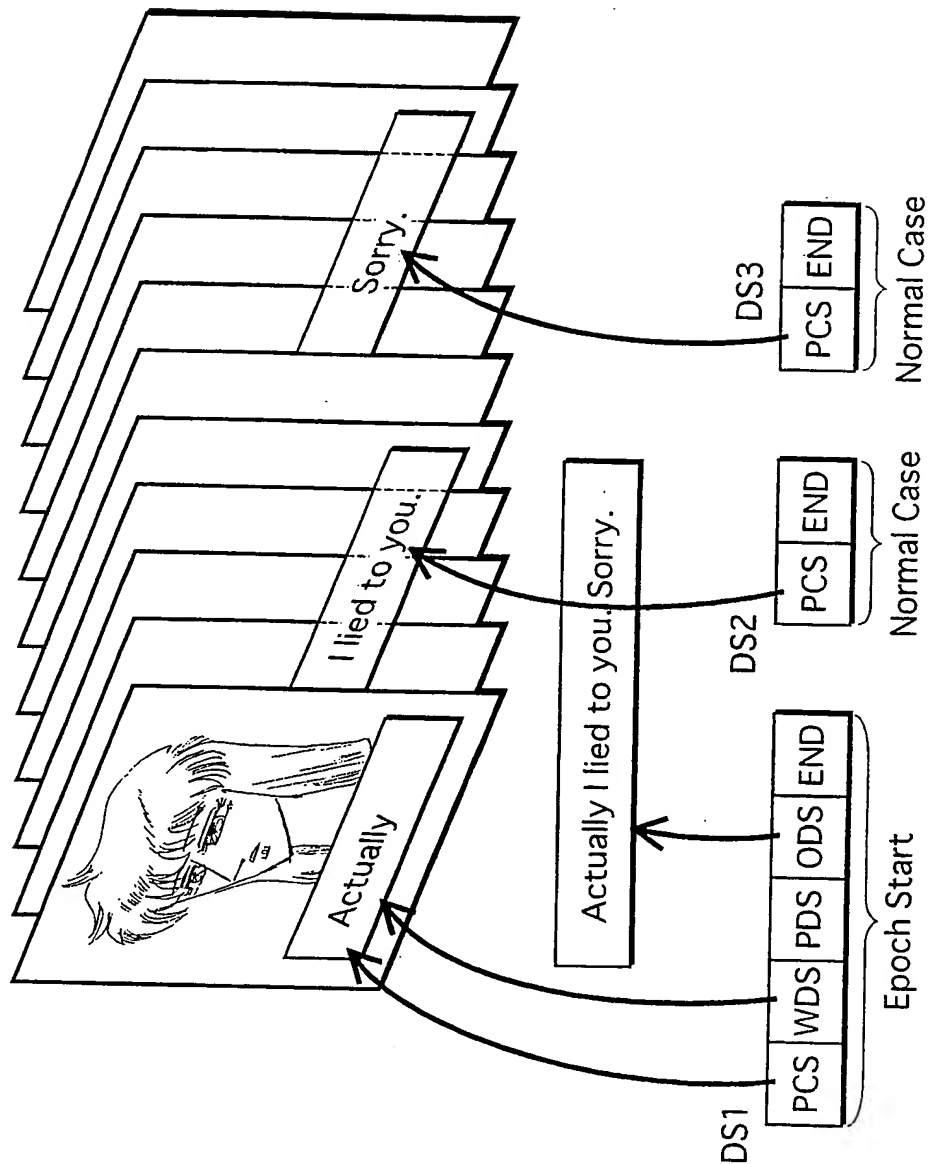


FIG.10

EXAMPLE DESCRIPTION
OF PCS AND WDS IN DS1

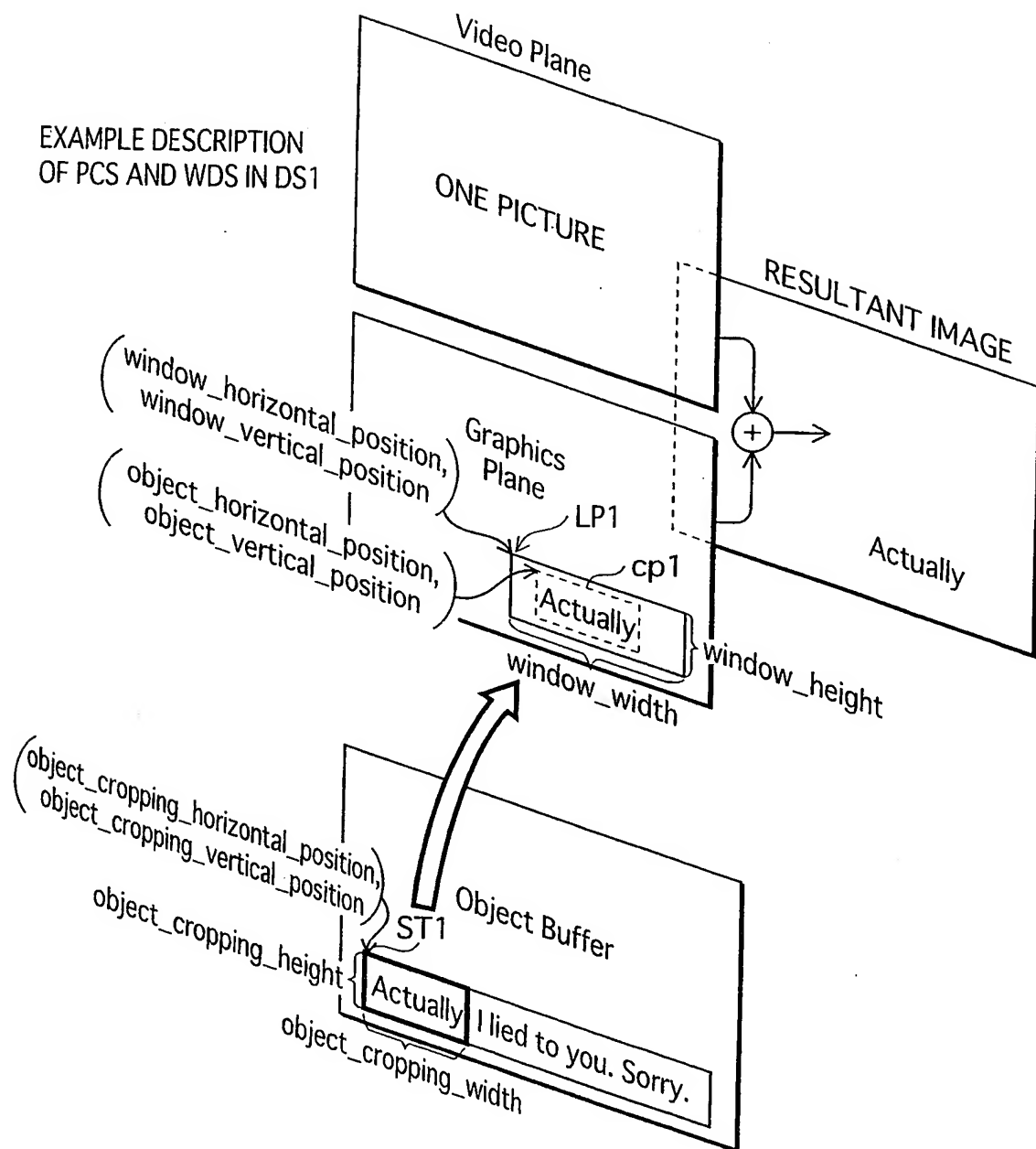


FIG. 11

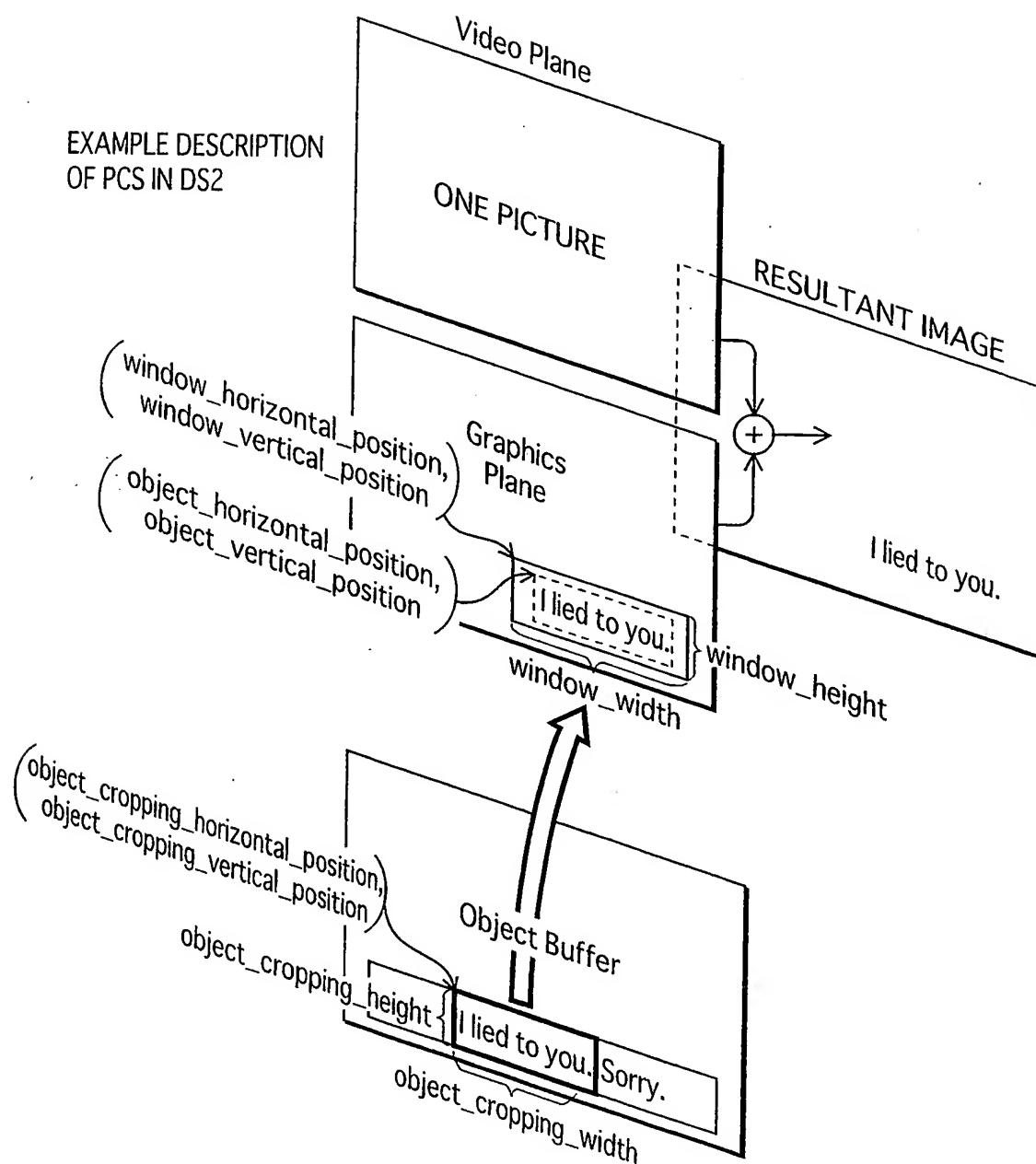


FIG.12

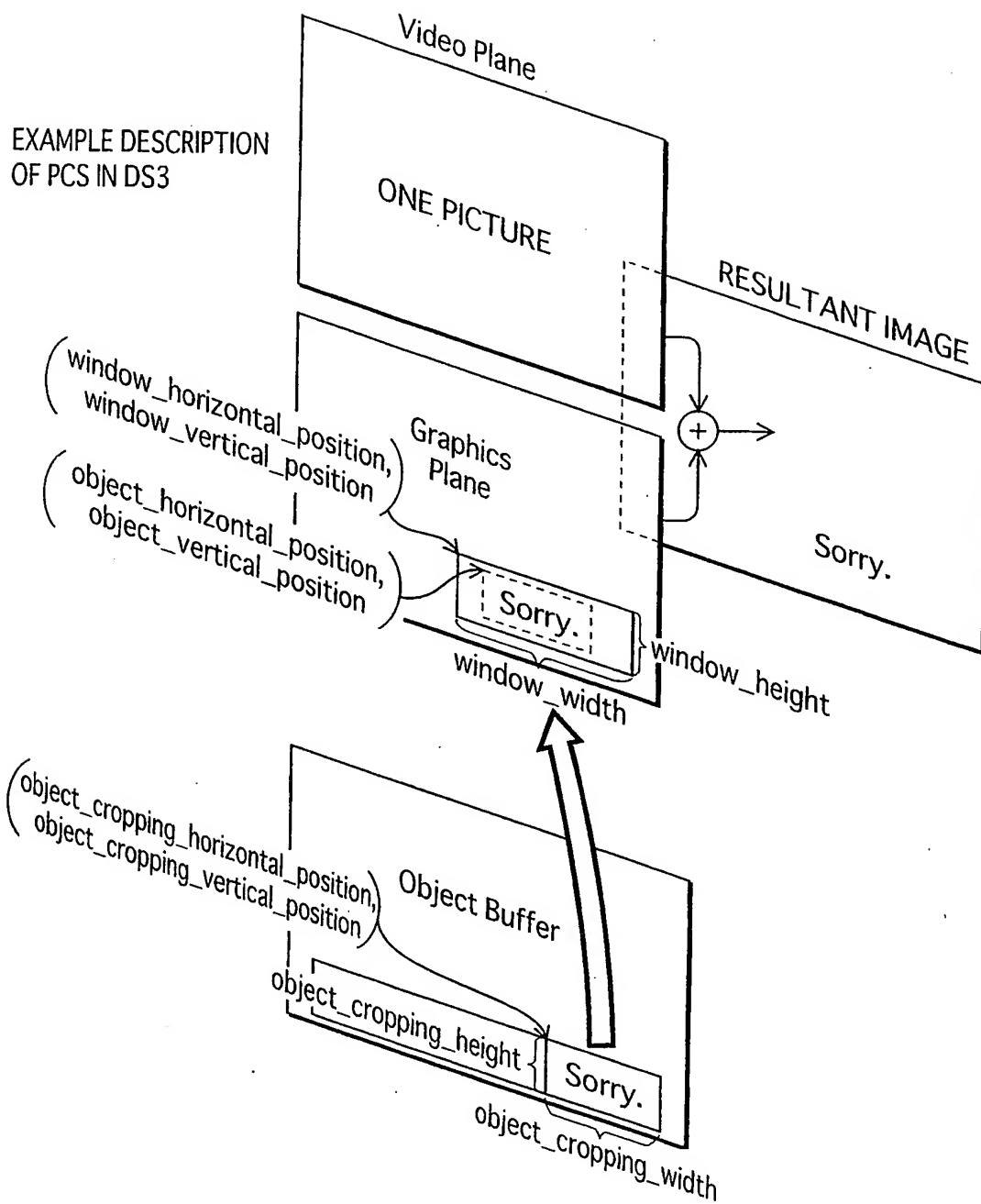


FIG.13

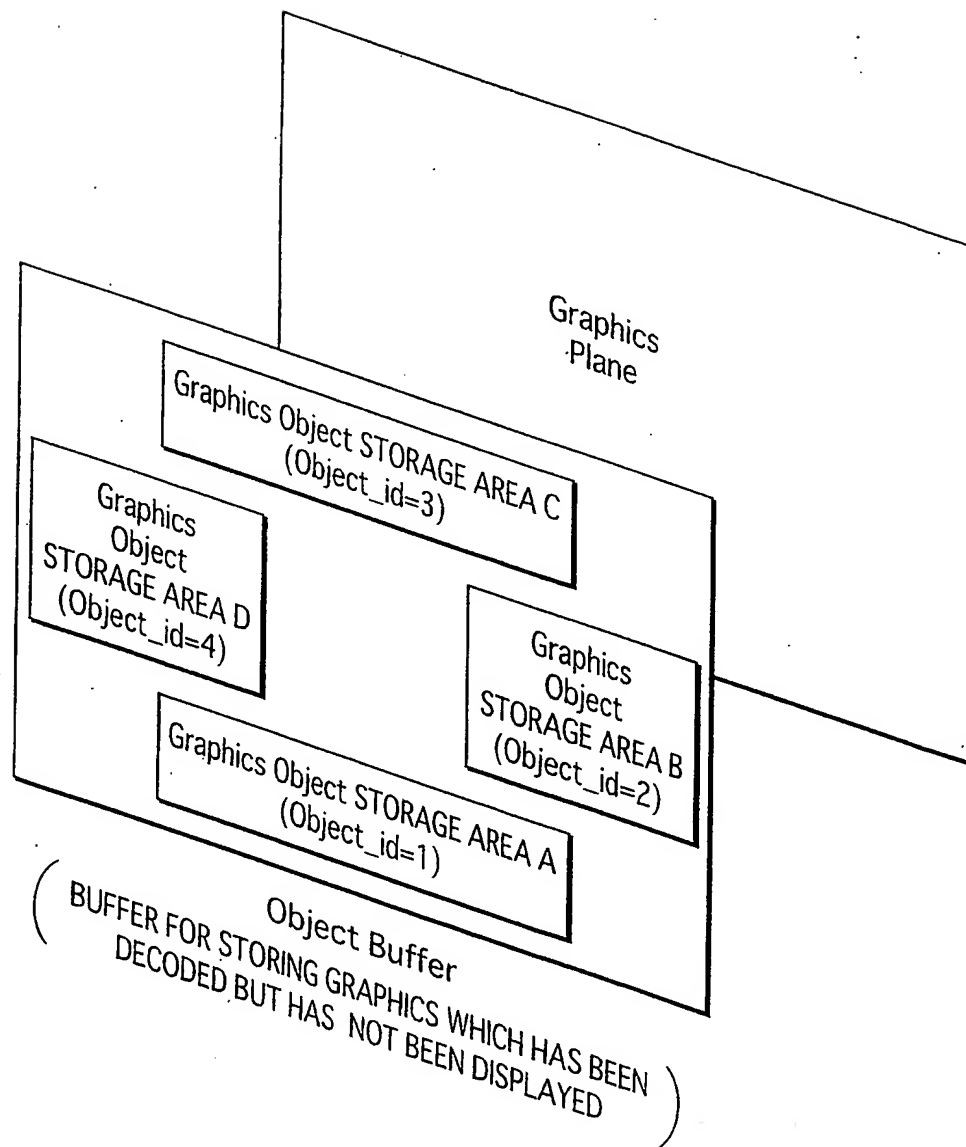


FIG. 14 $PTS(DSn[PCS]) \geq DTS(DSn[PCS]) + DECODEDURATION(DSn)$

Where:

- $DECODEDURATION(DSn)$ is calculated as follows:

```

decode_duration = 0 ;
decode_duration += PLANEINITIALIZATIONTIME( DSn ) ;
if( DSn. PCS. num_of_objects == 2 )
{
    decode_duration += WAIT( DSn, DSn. PCS. OBJ[0], decode_duration ) ;
    if( DSn. PCS. OBJ[0]. window_id == DSn. PCS. OBJ[1]. window_id )
    {
        decode_duration += WAIT( DSn, DSn. PCS. OBJ[1], decode_duration ) ;
        decode_duration += 90000*( SIZE( DSn. PCS. OBJ[0]. window_id )//256*106 ) ;
    }
    else
    {
        decode_duration += 90000*( SIZE( DSn. PCS. OBJ[0]. window_id )//256*106 ) ;
        decode_duration += WAIT( DSn, DSn. PCS. OBJ[1], decode_duration ) ;
        decode_duration += 90000*( SIZE( DSn. PCS. OBJ[1]. window_id )//256*106 ) ;
    }
}
else if( DSn. PCS. num_of_objects == 1 )
{
    decode_duration += WAIT( DSn, DSn. PCS. OBJ[0], decode_duration ) ;
    decode_duration += 90000*( SIZE( DSn. PCS. OBJ[0]. window_id )//256*106 ) ;
}
return decode_duration ;

```
- $PLANEINITIALIZATIONTIME(DSn)$ is calculated as follows:

```

initialize_duration=0 ;
if( DSn. PCS. composition_state== EPOCH_START )
{
    initialize_duration = 90000*( 8*video_width*video_height//256*106 ) ;
}
else
{
    for( i=0 ; i < WDS. num_windows ; i++ )
    {
        if( EMPTY(DSn.WDS.WIN[i],DSn) )
            initialize_duration += 90000*( SIZE( DSn. WDS. WIN[i] )//256*106 ) ;
    }
}
return initialize_duration ;

```
- $WAIT(DSn, OBJ, current_duration)$ is calculated as follows:

```

wait_duration = 0 ;
if( EXISTS( OBJ. object_id, DSn ) )
{
    object_definition_ready_time = PTS( GET( OBJ. object_id. DSn ) ) ;
    current_time = DTS( DSn. PCS )+current_duration ;
    if( current_time < object_definition_ready_time )
        wait_duration += object_definition_ready_time - current_time ;
}
return wait_duration ;

```

FIG. 15

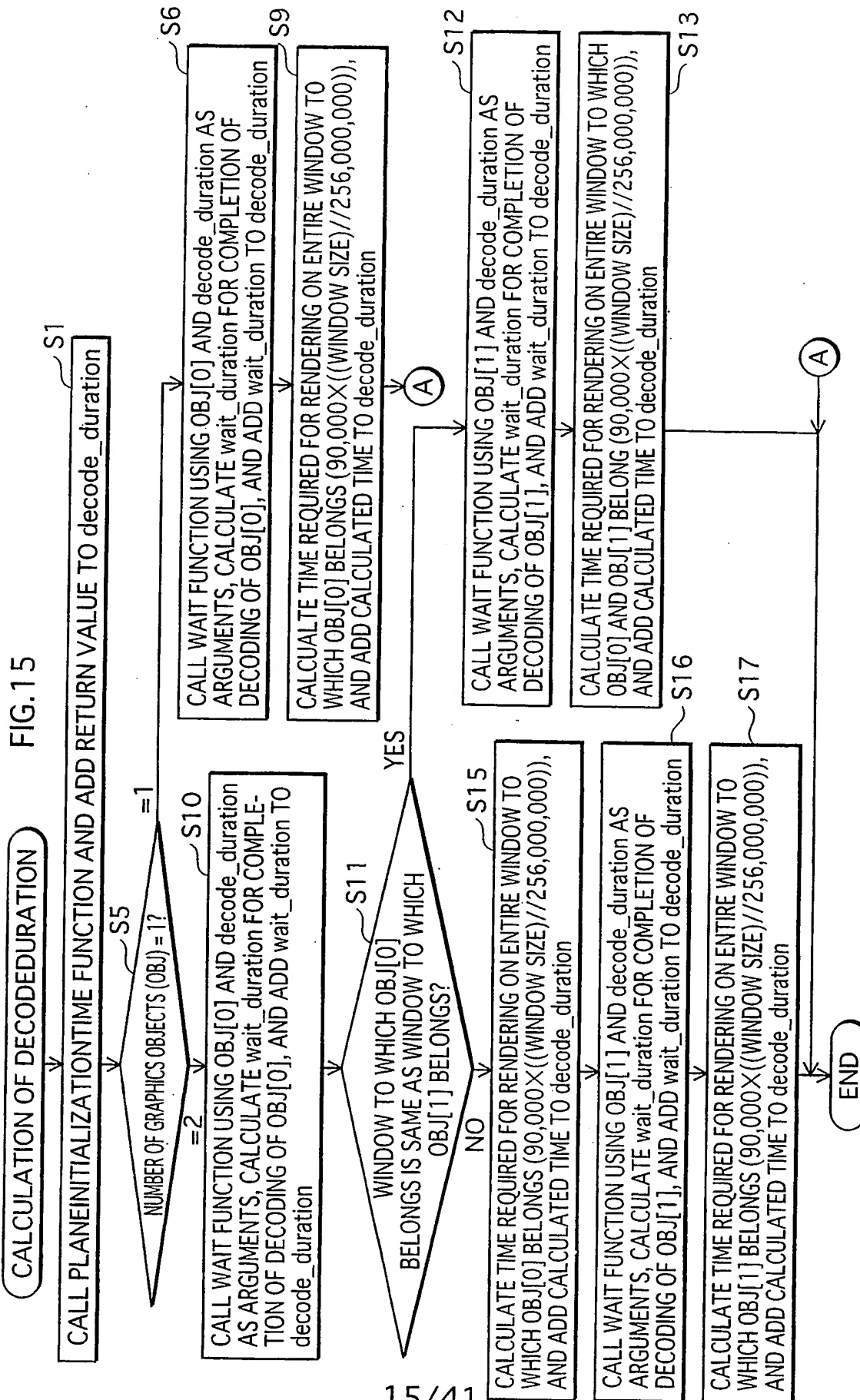


FIG.16A

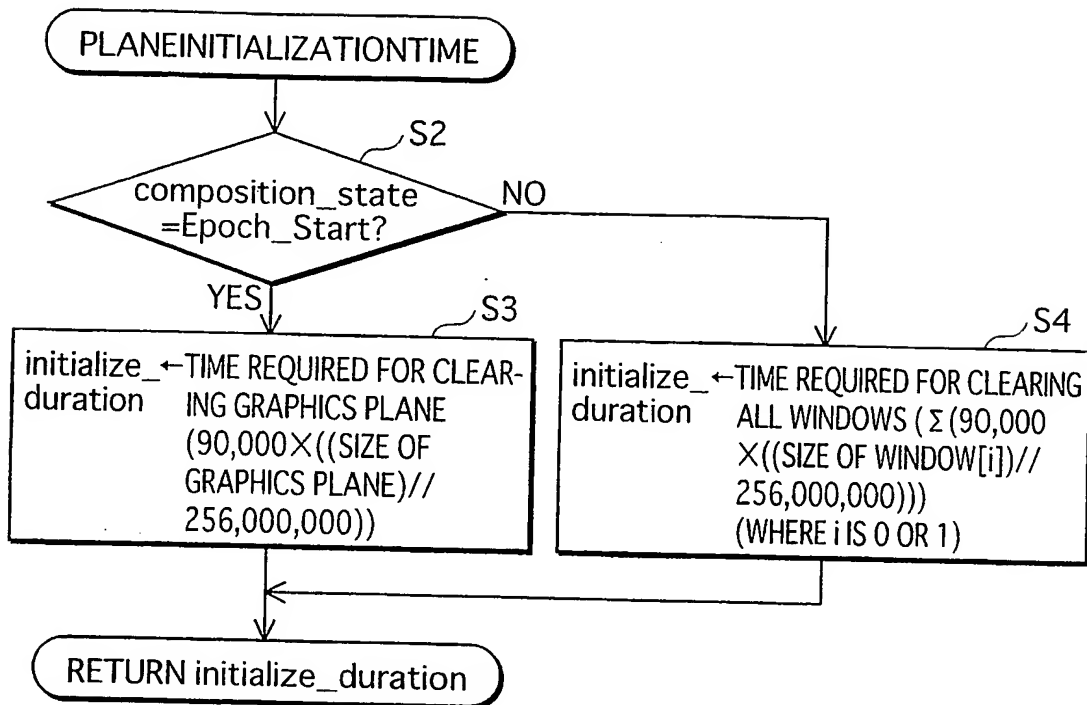


FIG.16B

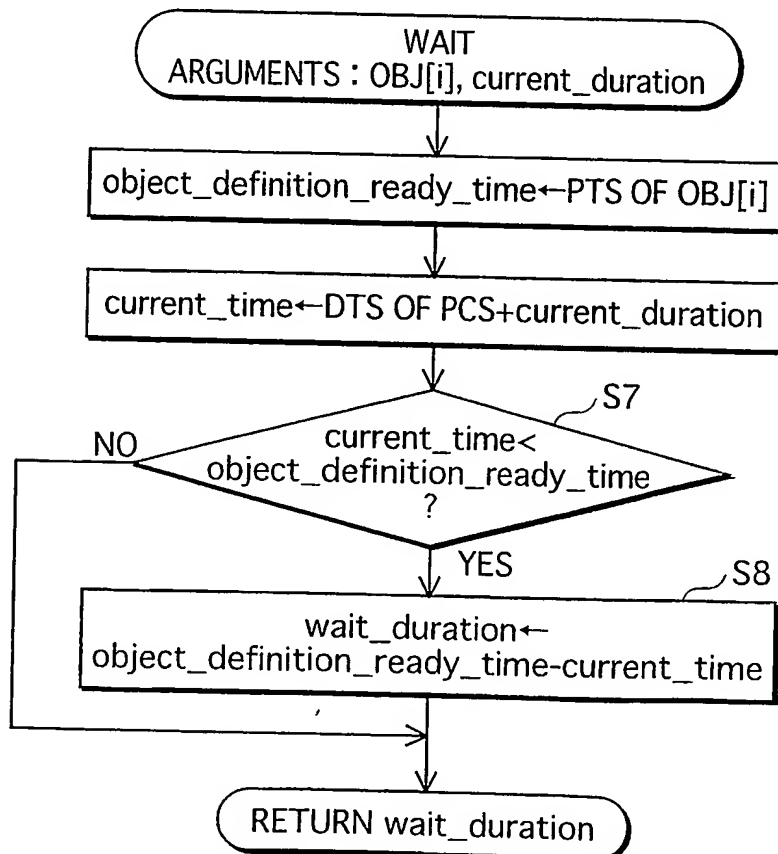


FIG.17A

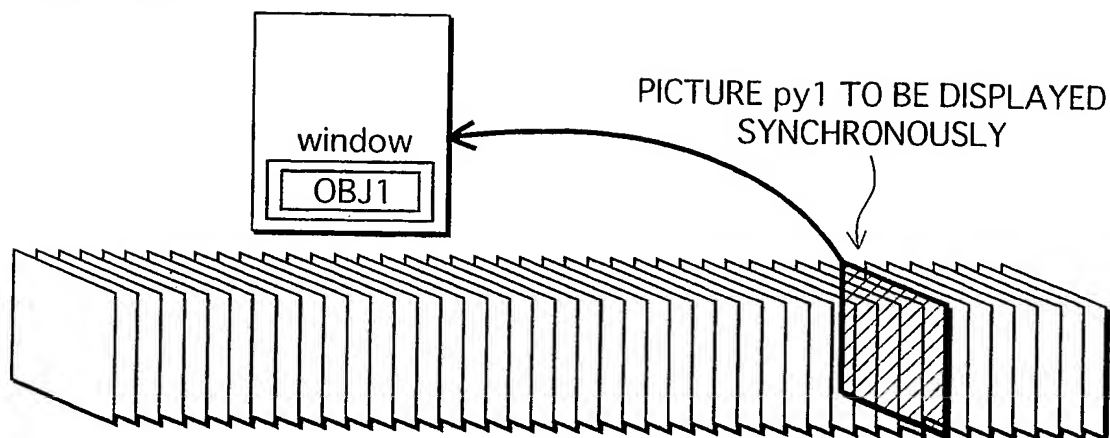


FIG.17B

DECODE_DURATION
 $= (2) + (3)$

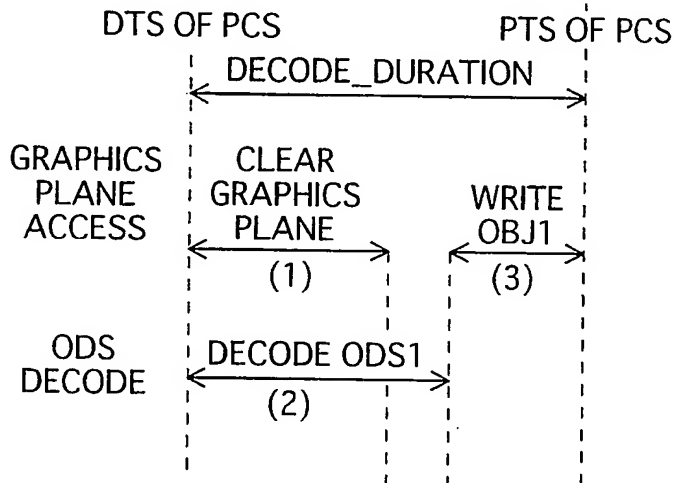


FIG.17C

DECODE_DURATION
 $= (1) + (3)$

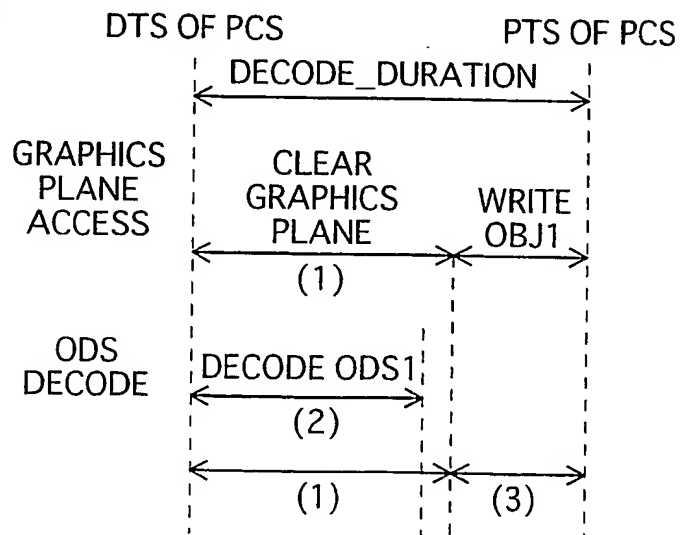


FIG.18A

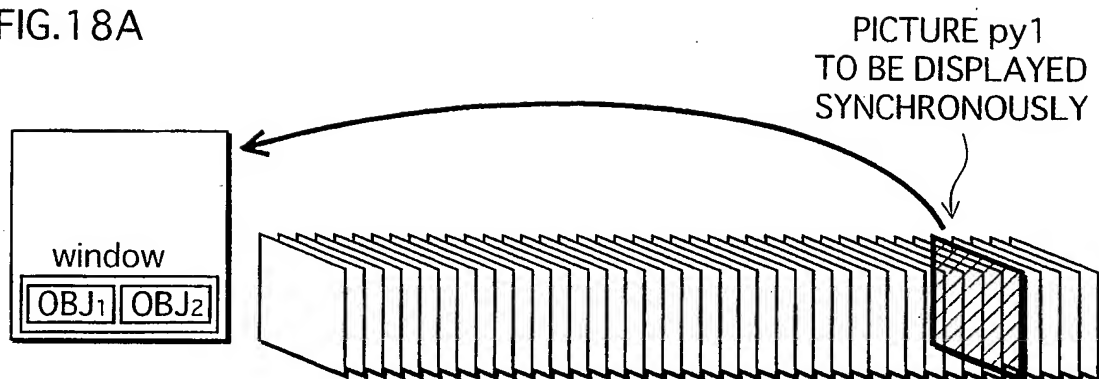


FIG.18B

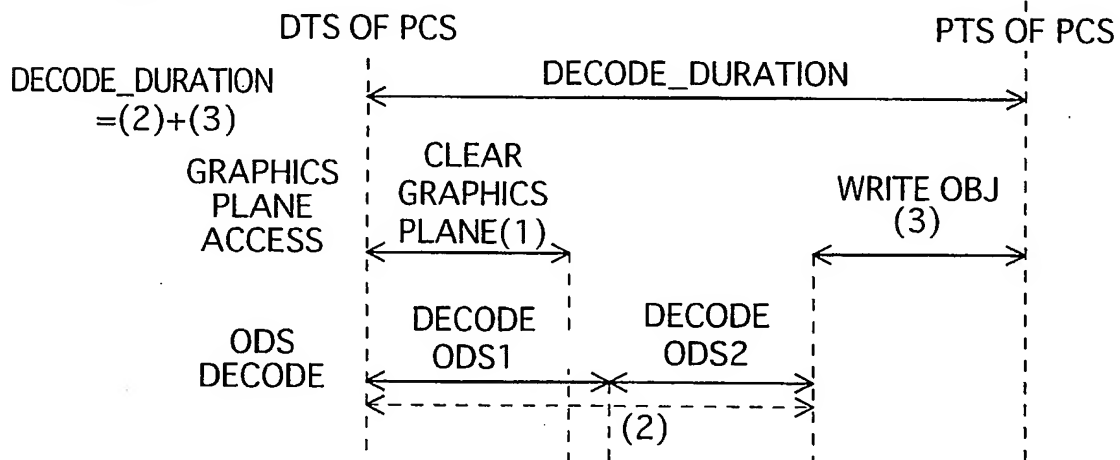
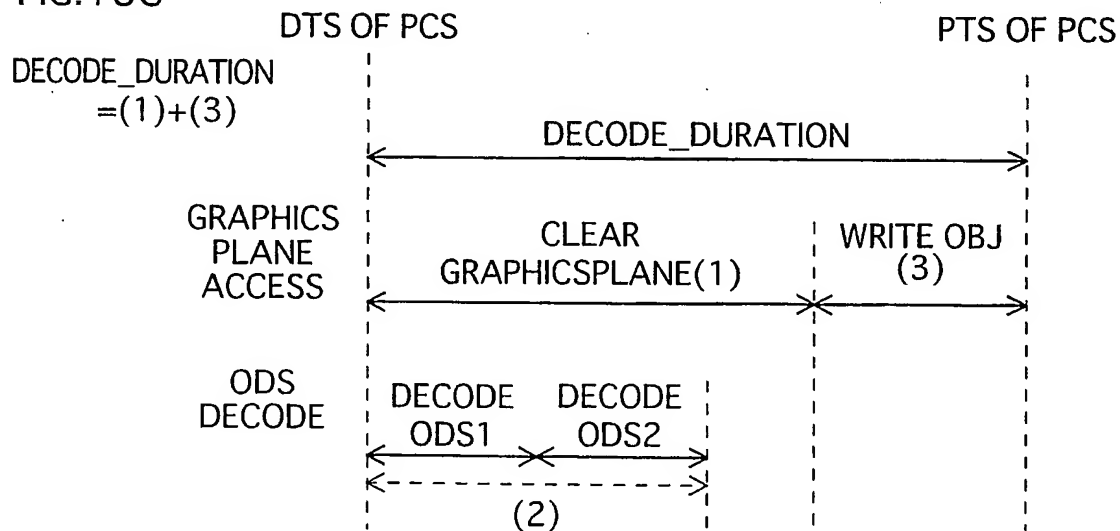


FIG.18C



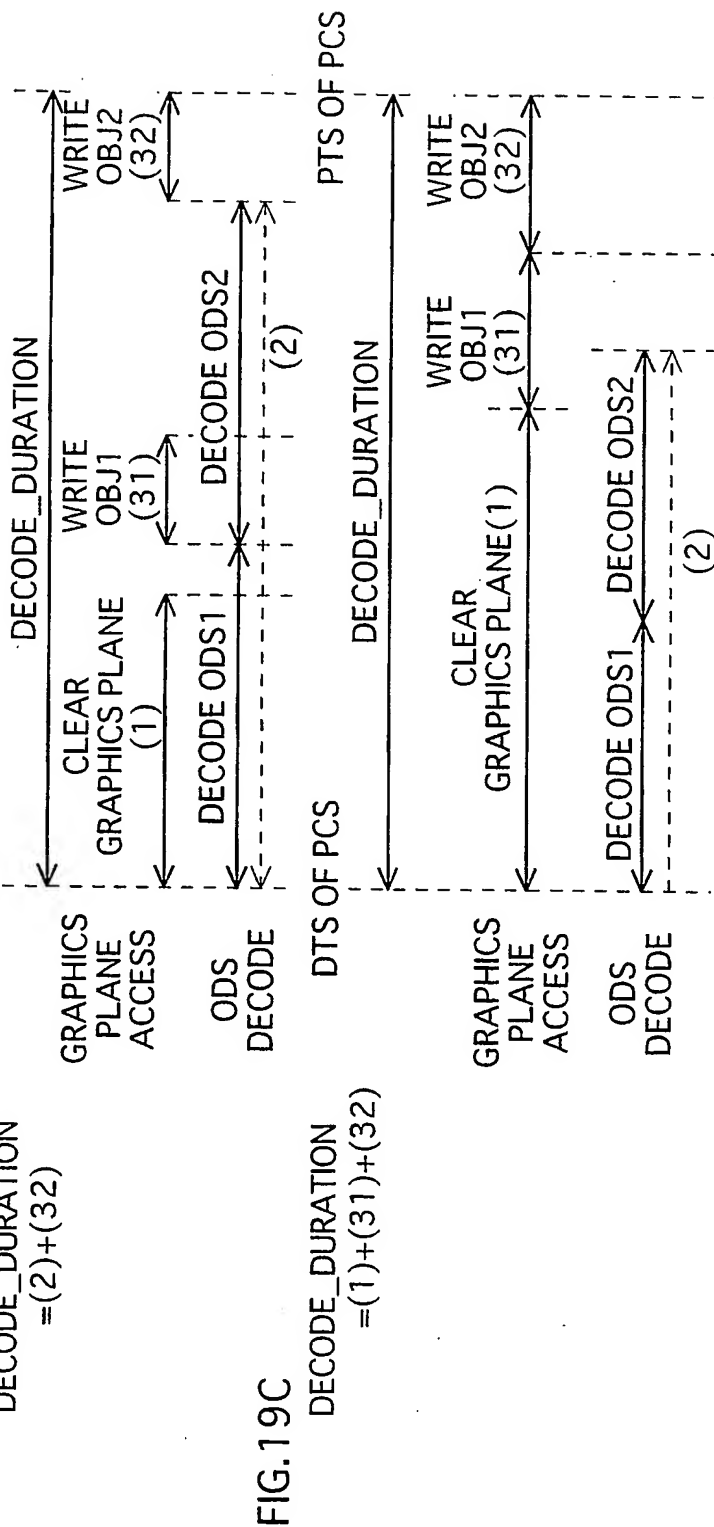
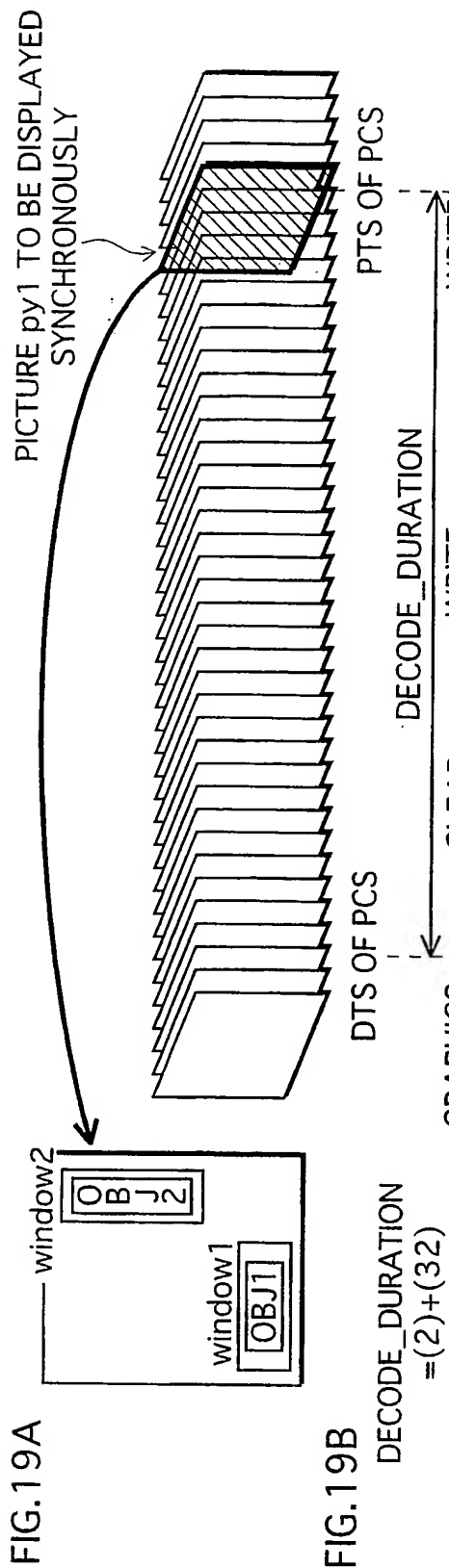


FIG.20

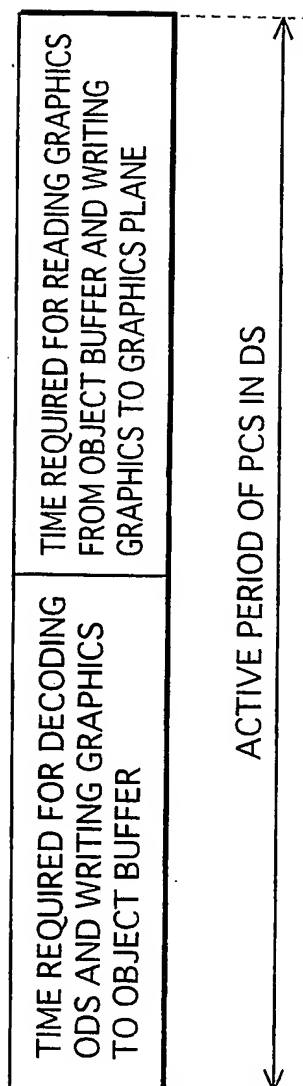


FIG.21

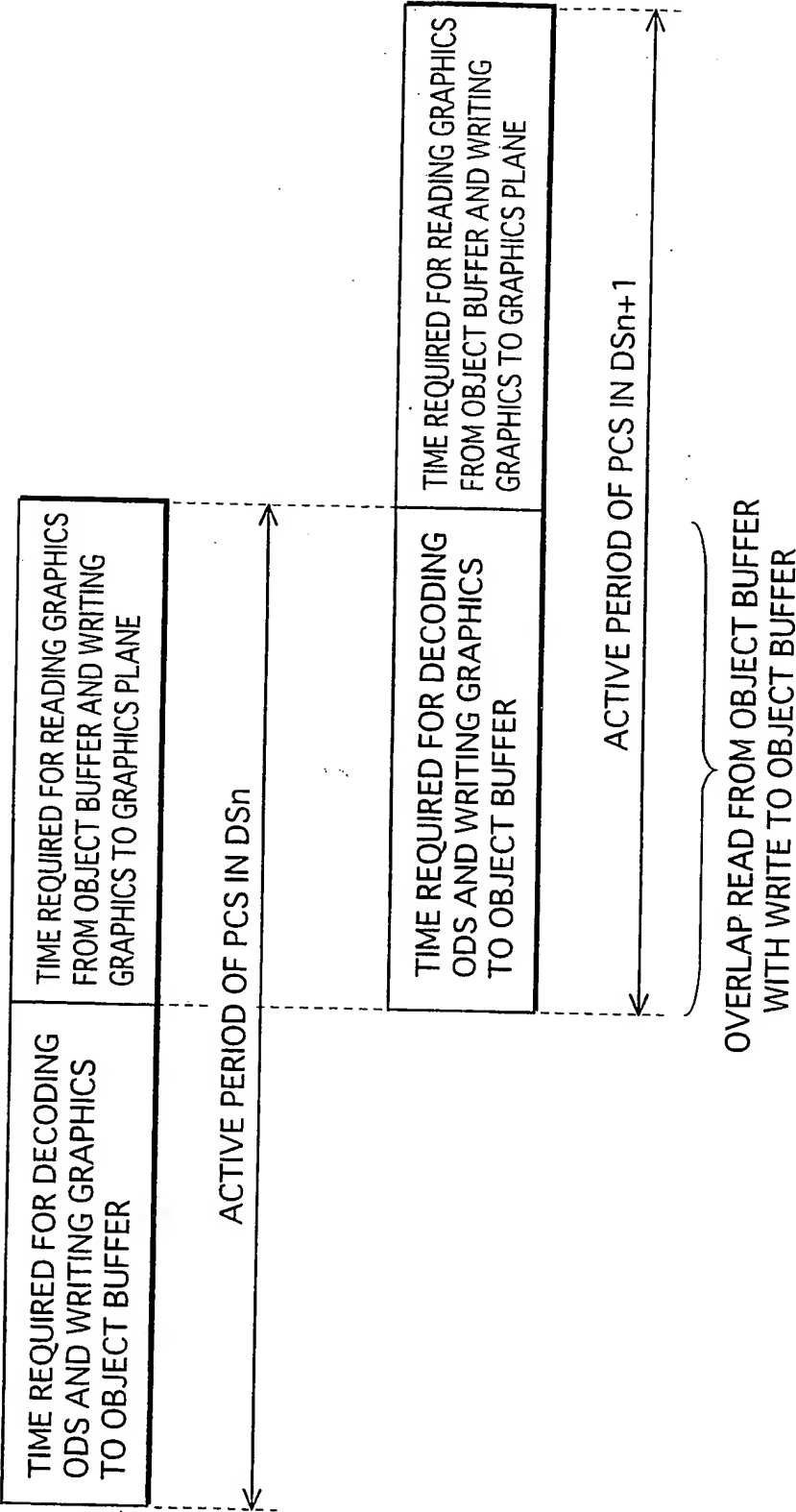


FIG.22

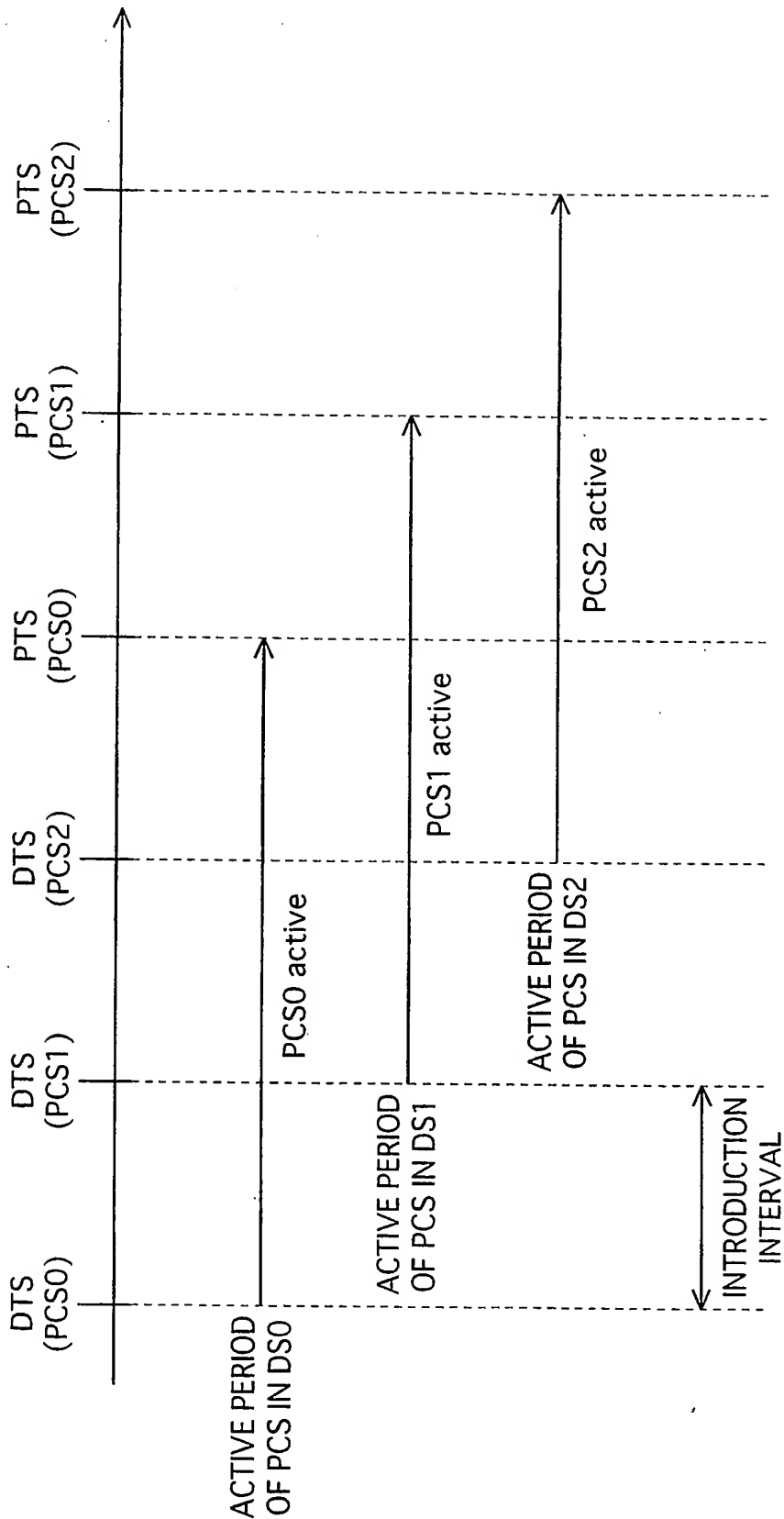


FIG. 23

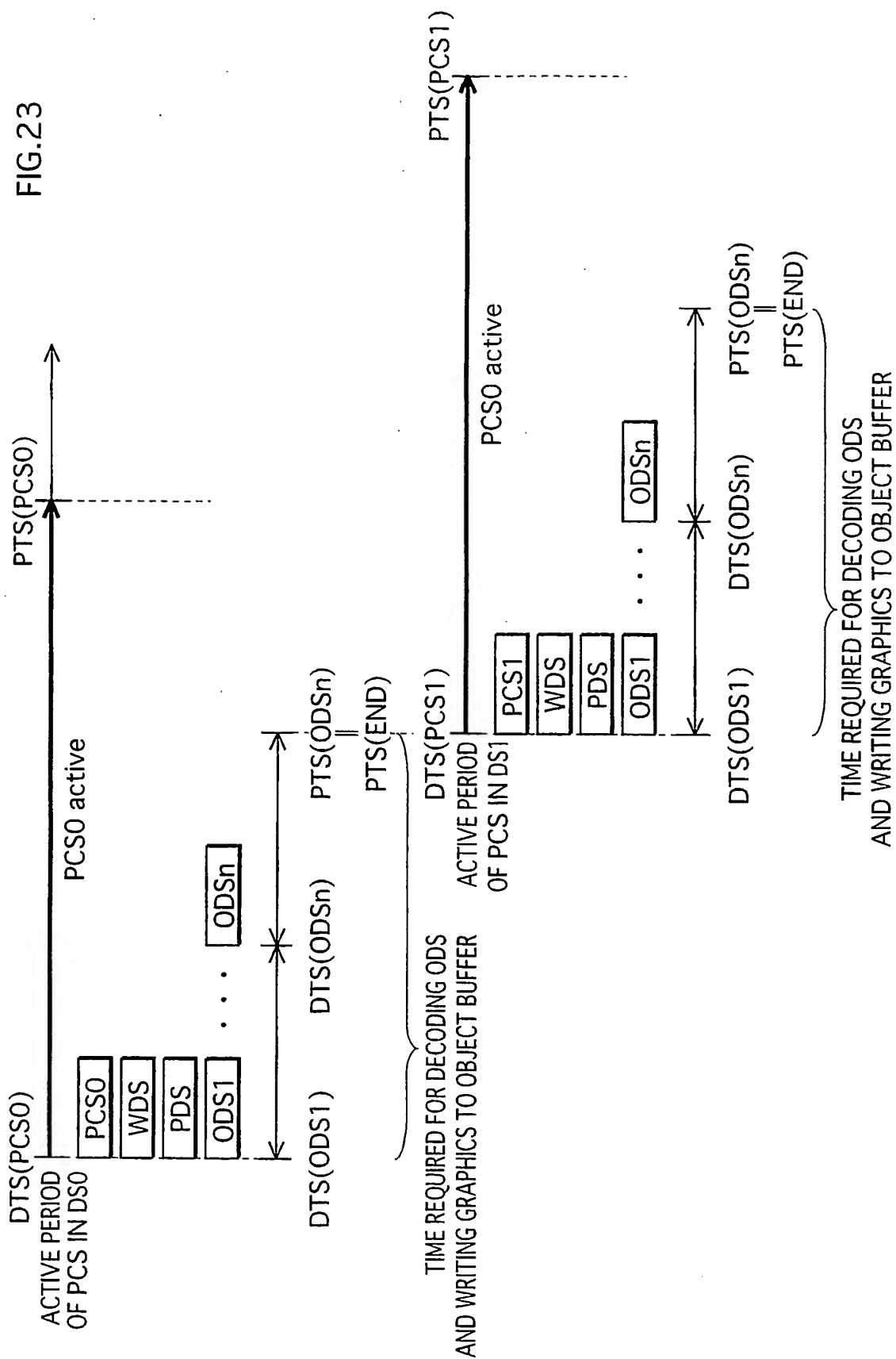


FIG.24

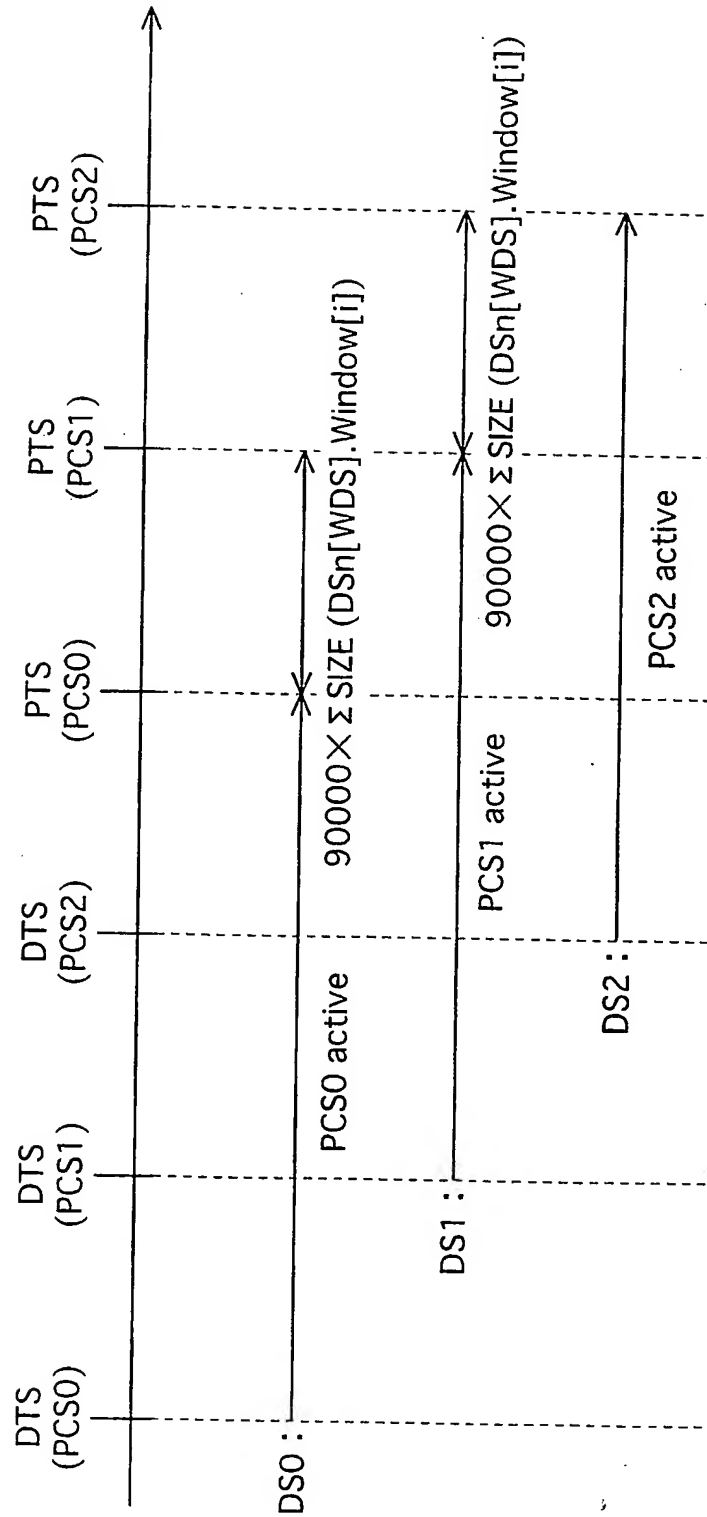


FIG.25A PIPELINE

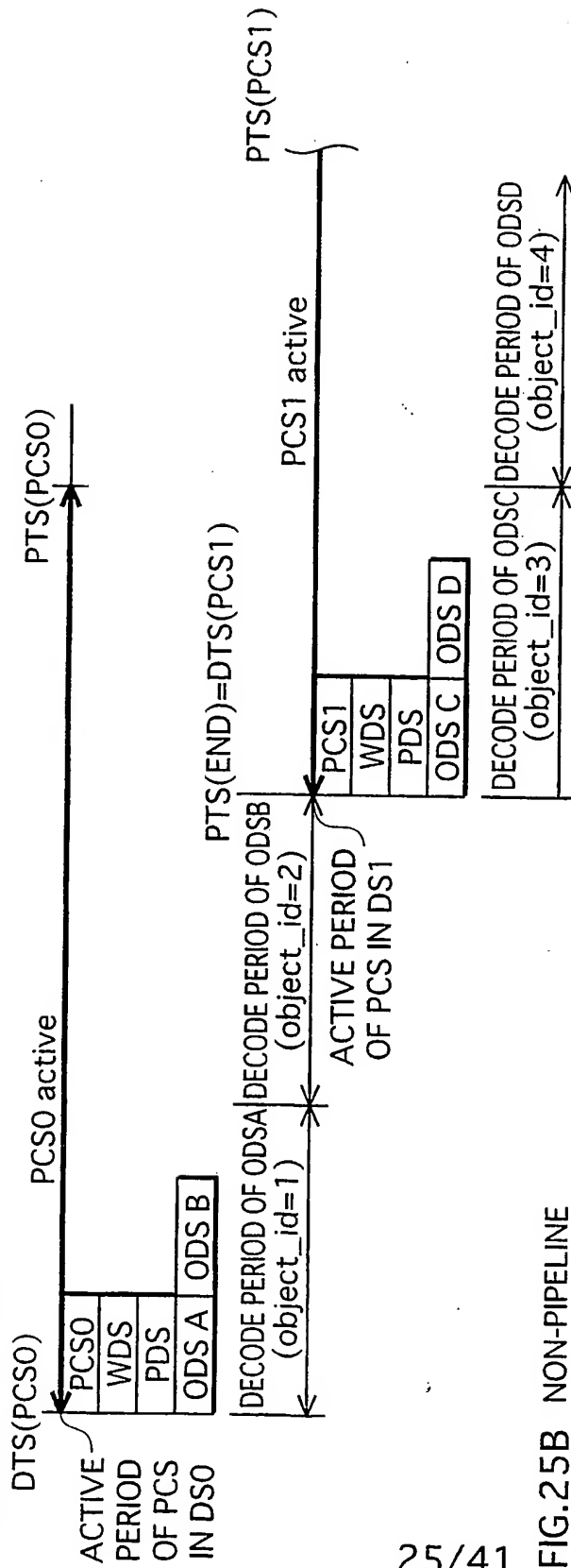


FIG.25B NON-PIPELINE

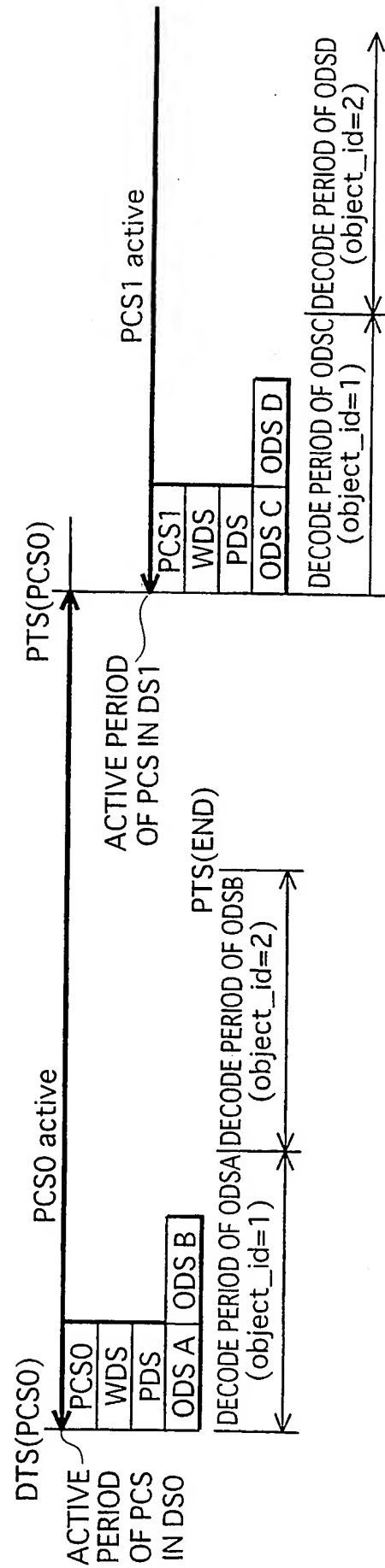


FIG.26

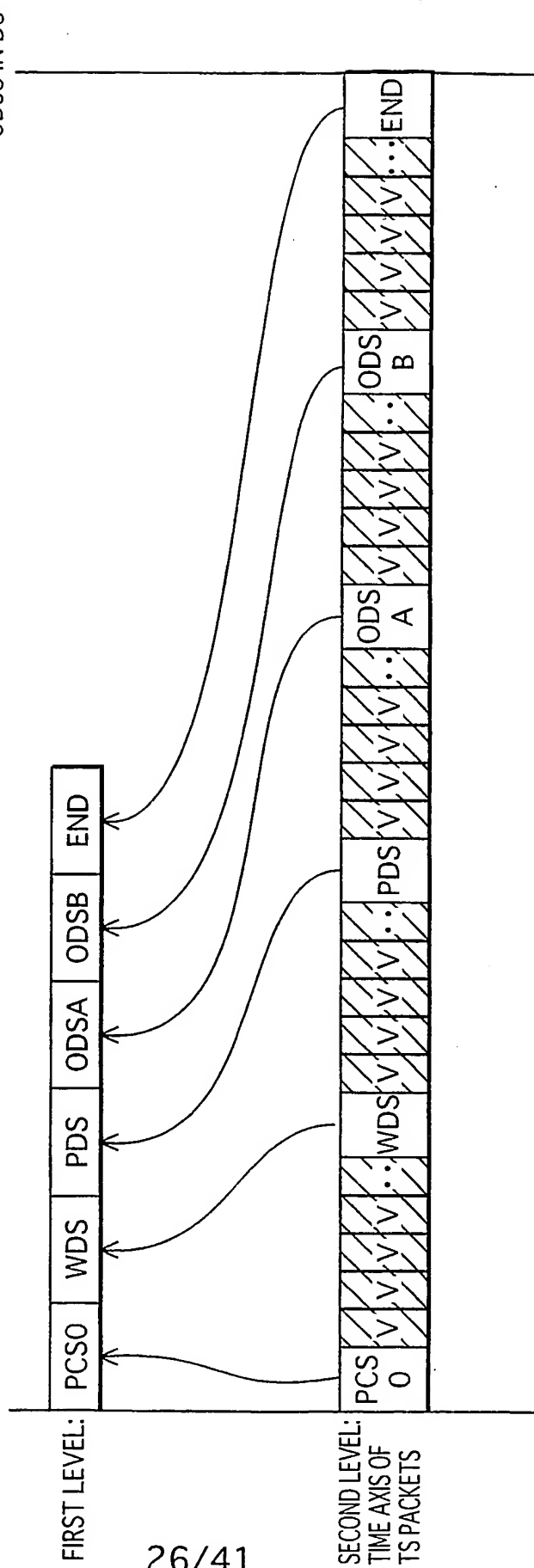


FIG.27A SCREEN COMPOSITION

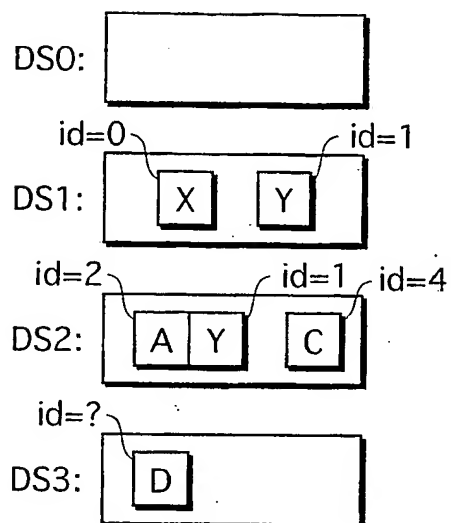


FIG.27B ACTIVE PERIOD OVERLAPPING AND ODS TRANSFER

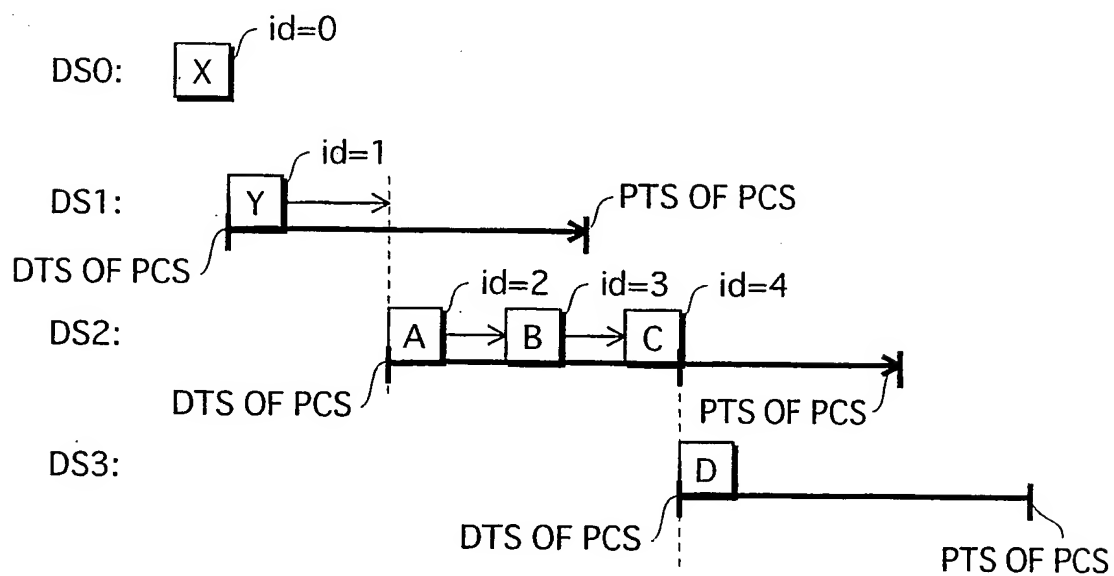


FIG.27C ARRANGEMENT IN OBJECT BUFFER

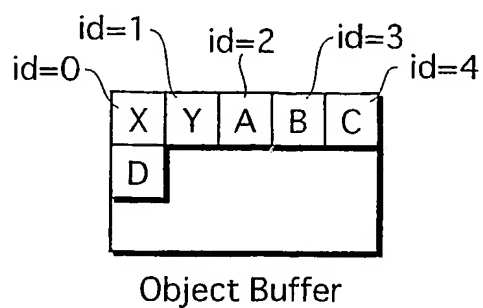


FIG.28

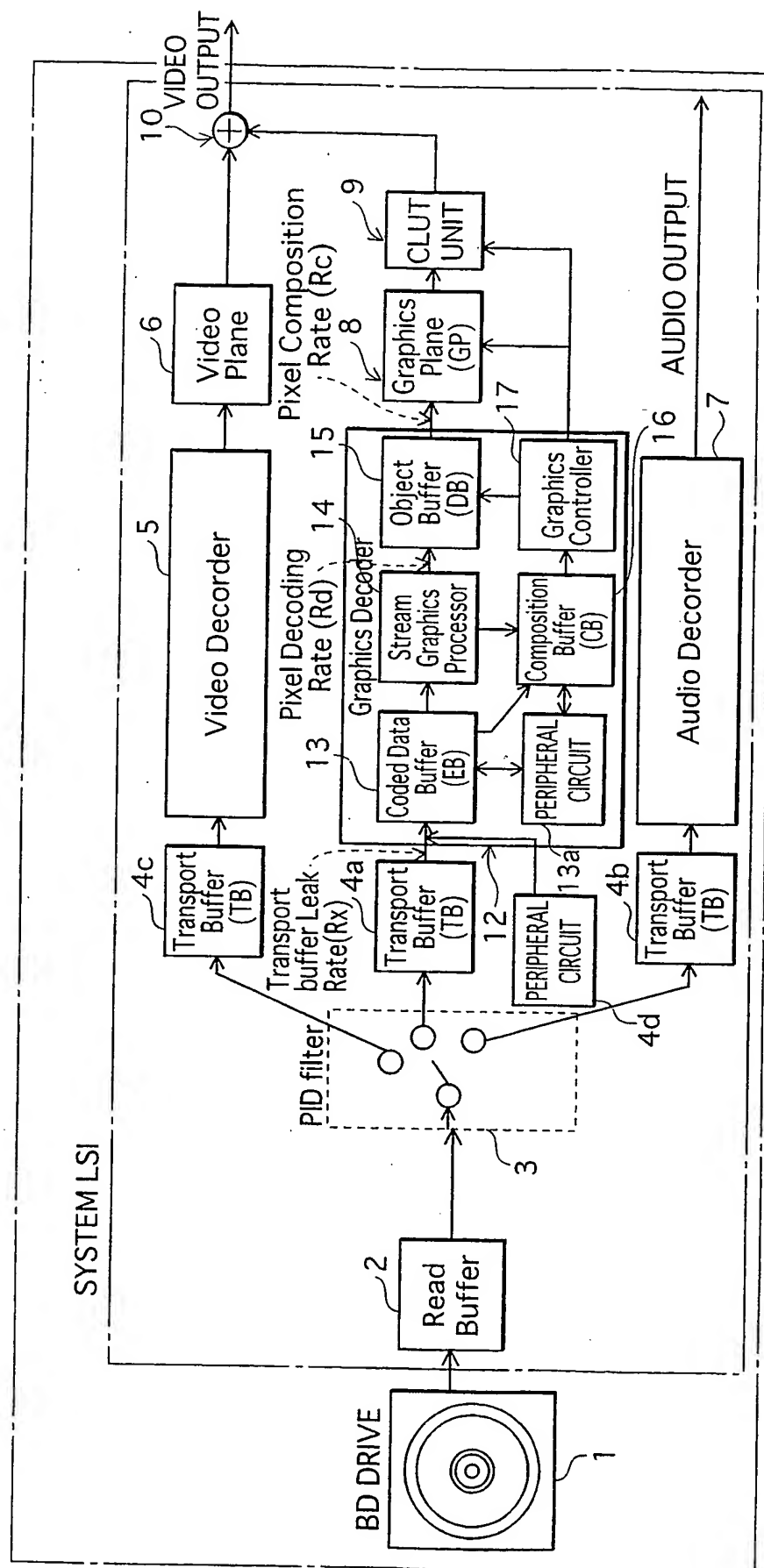
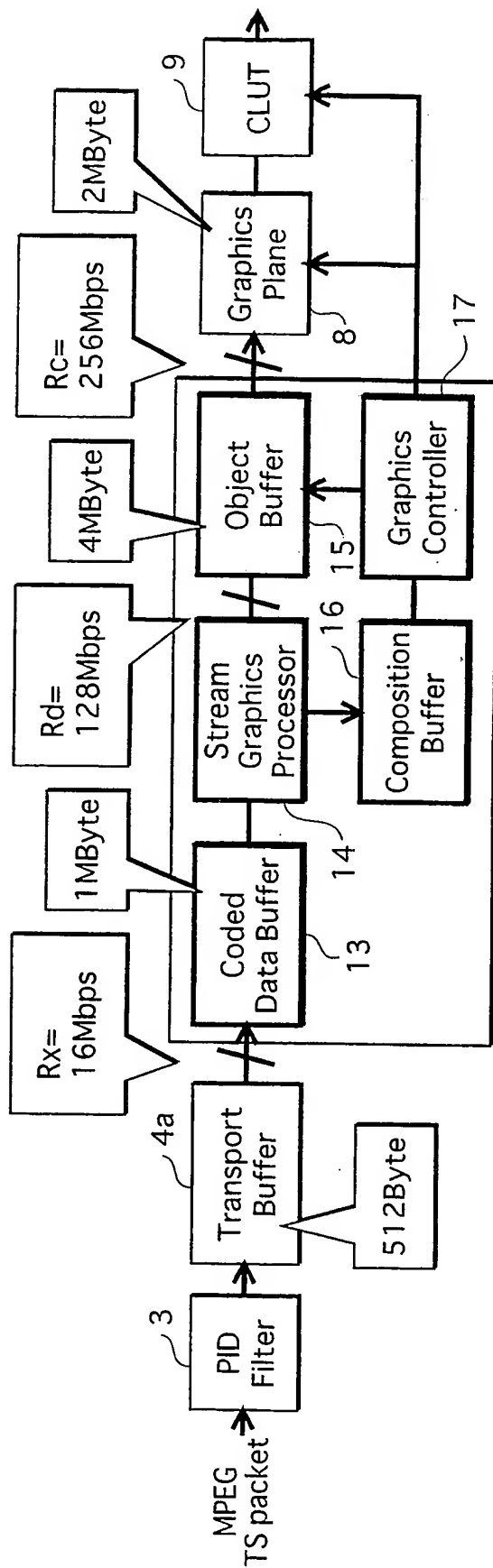
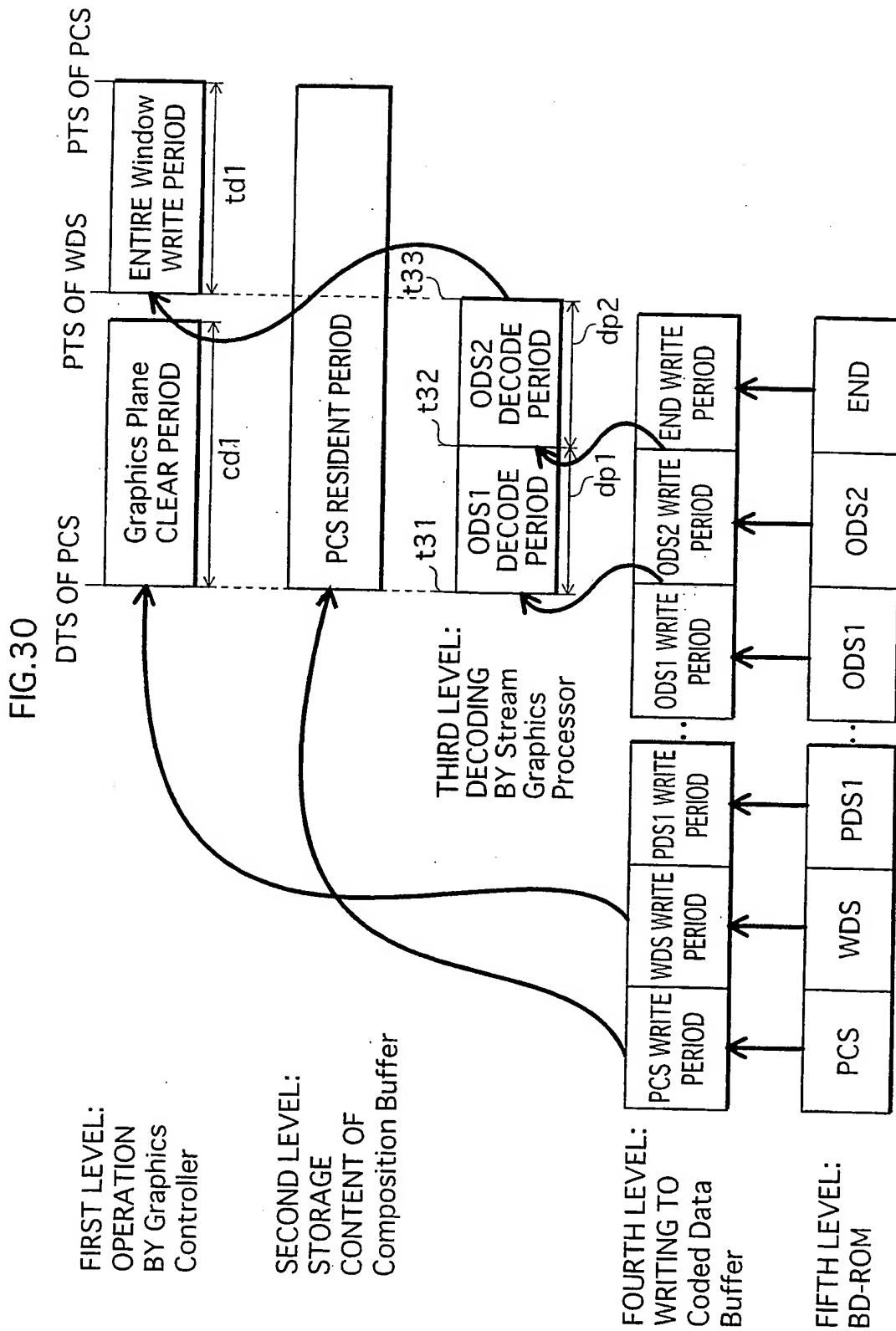
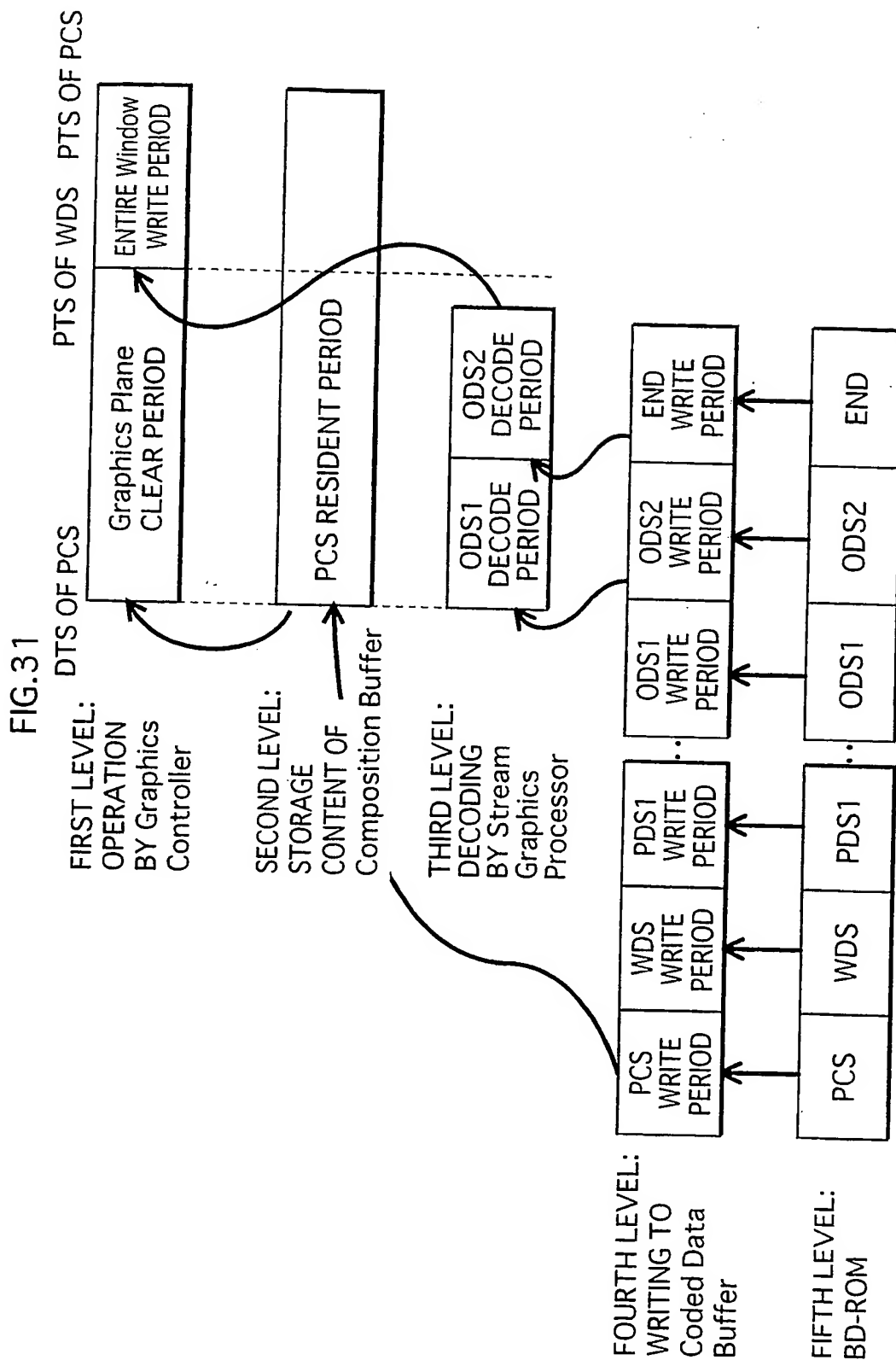


FIG.29







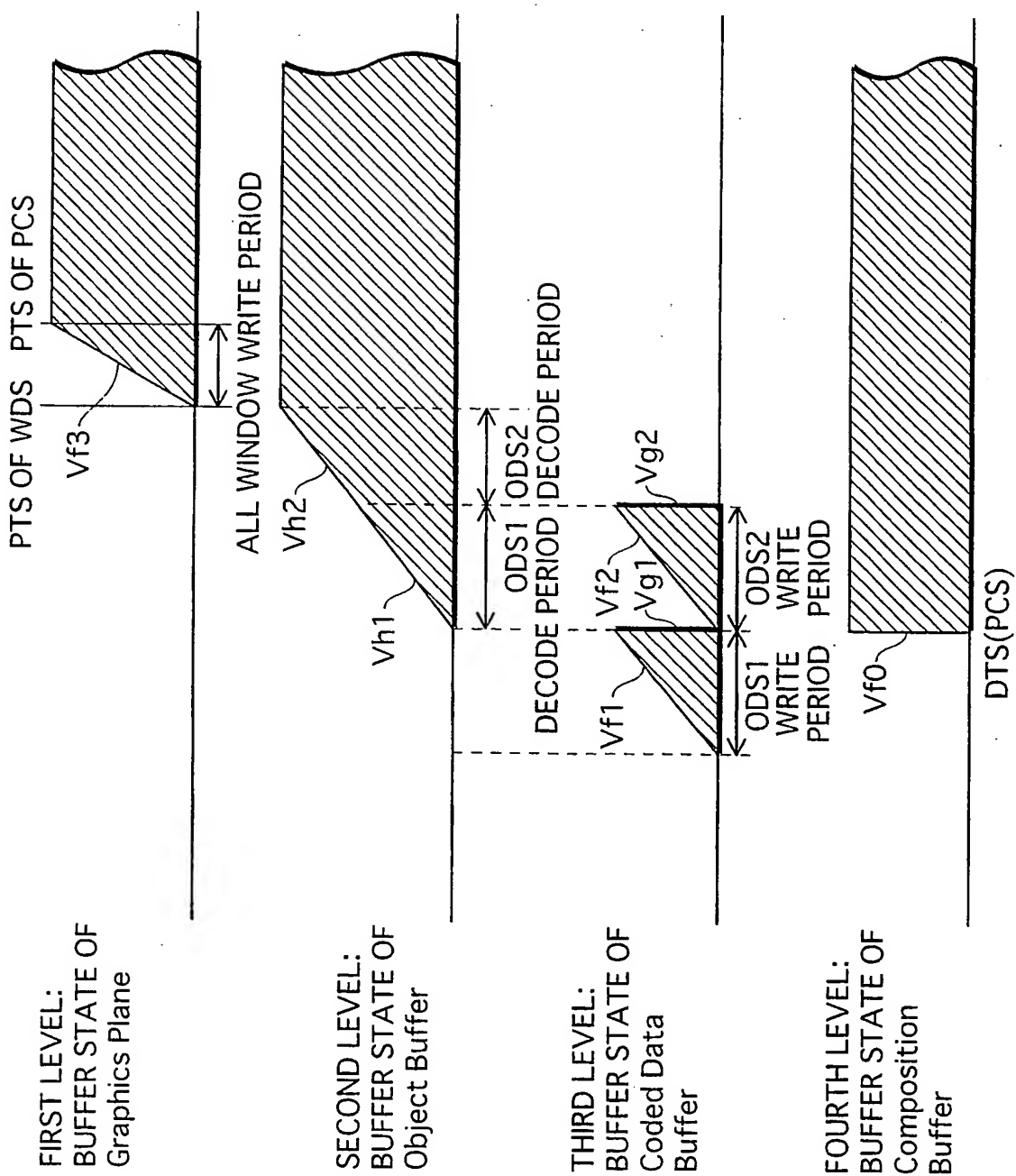


FIG. 33

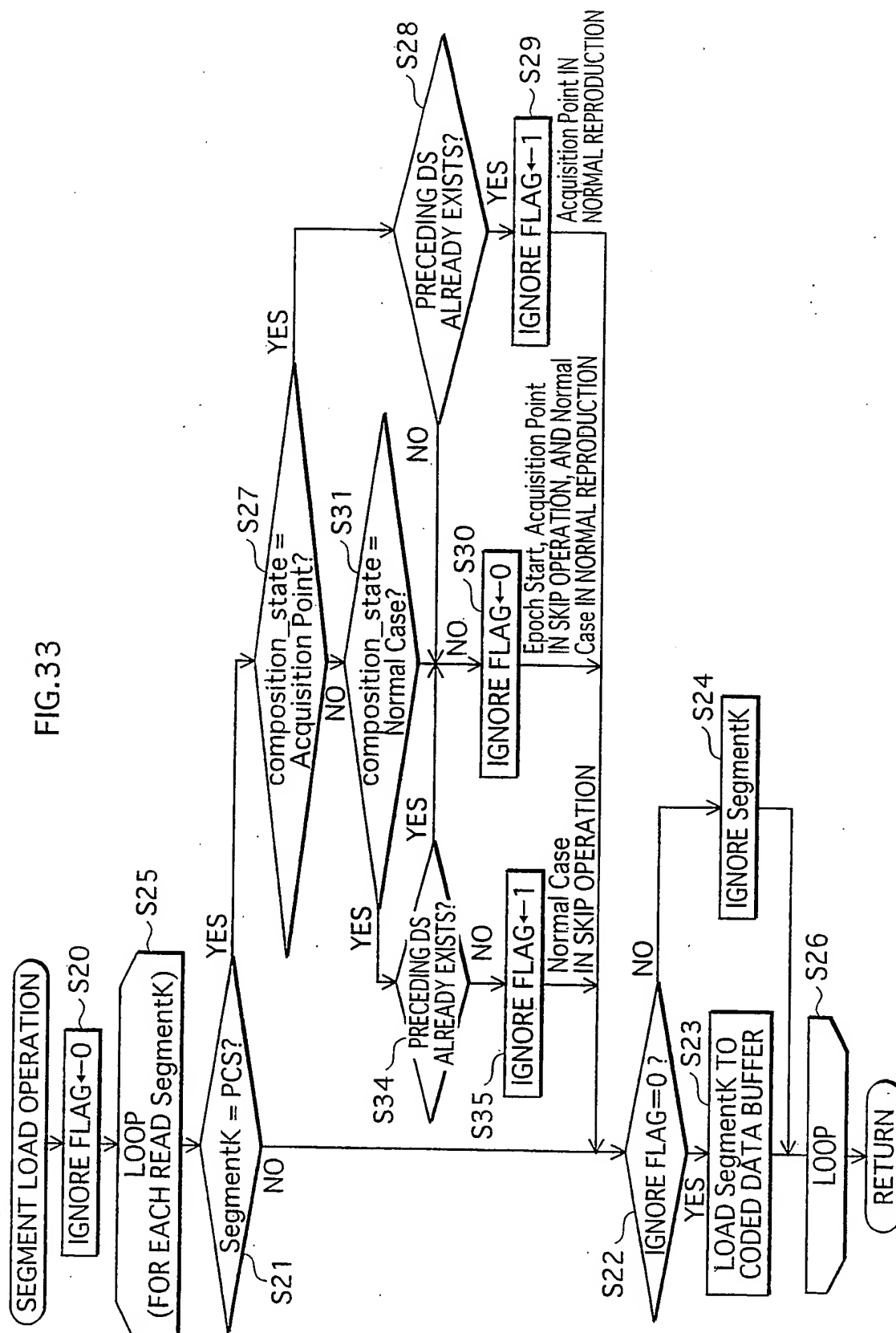


FIG.34

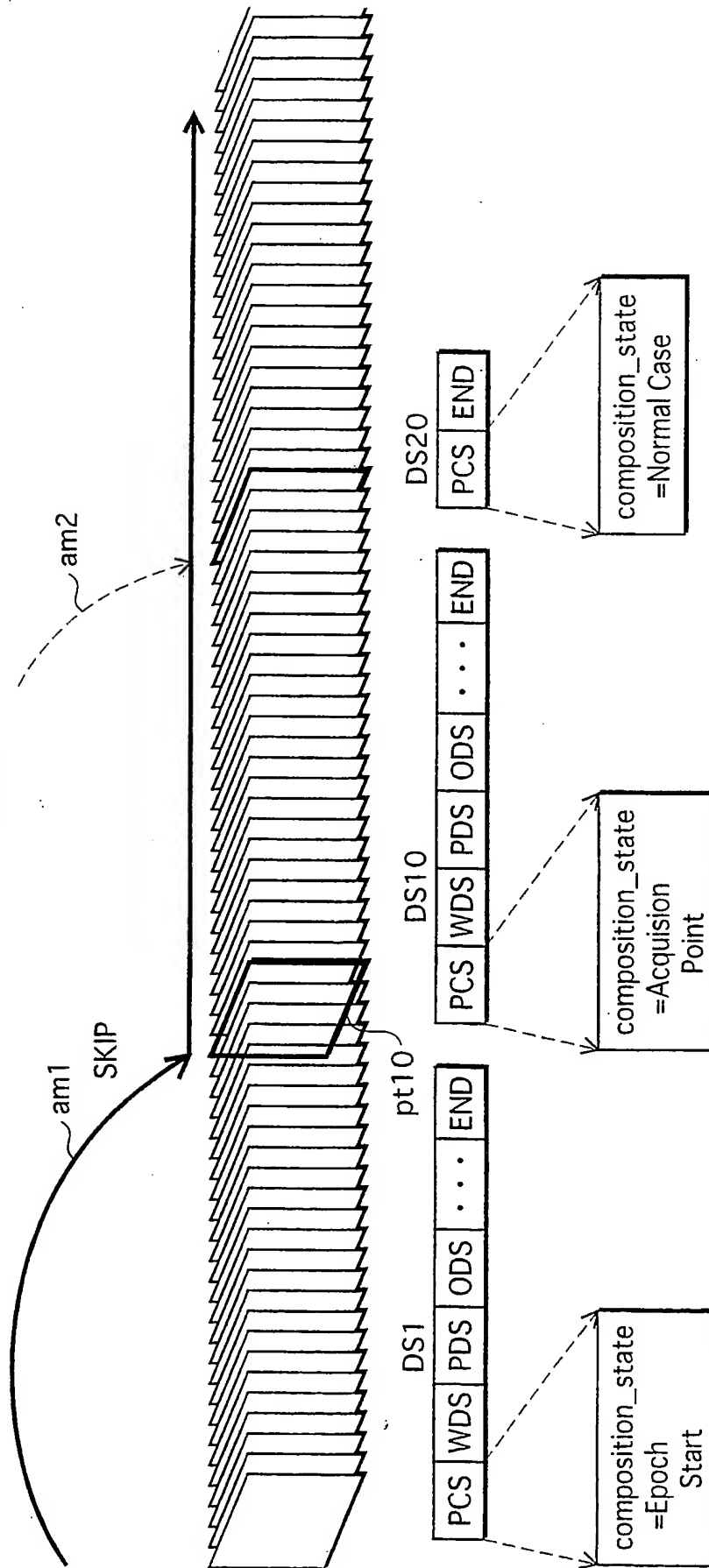


FIG.35

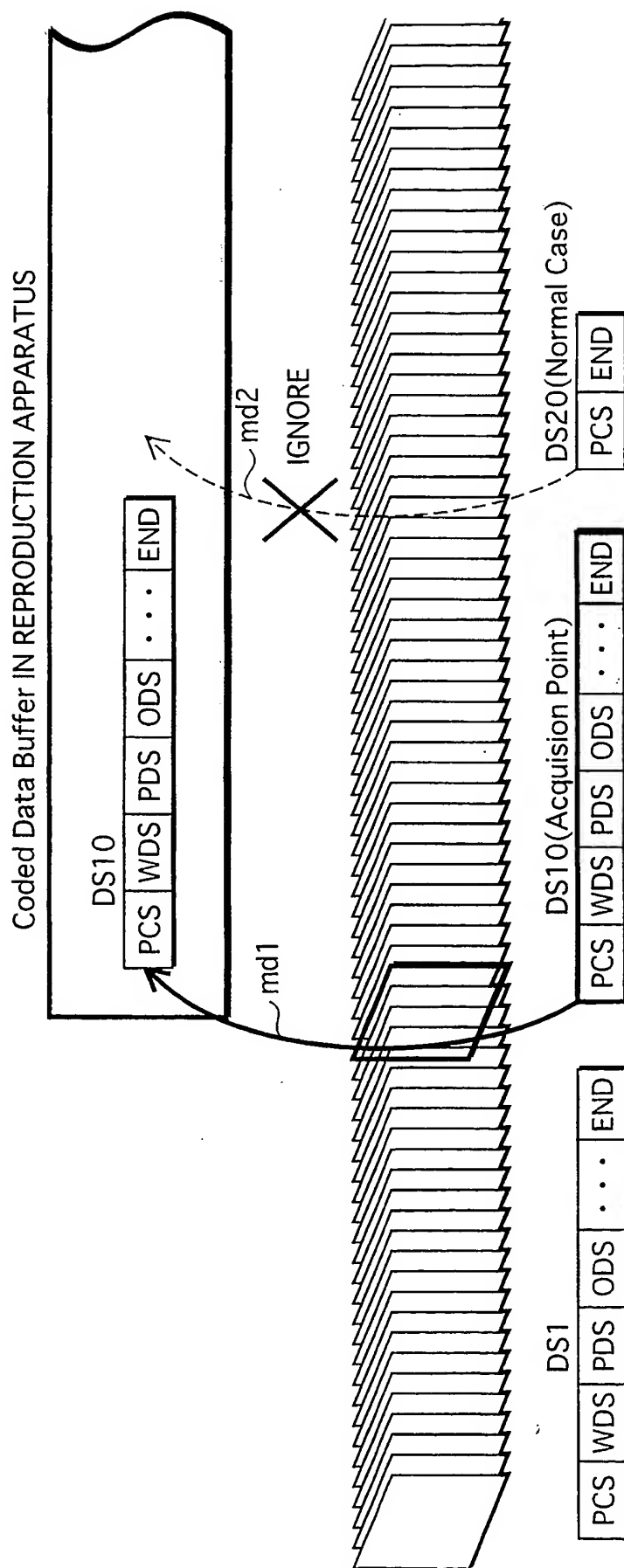


FIG.36

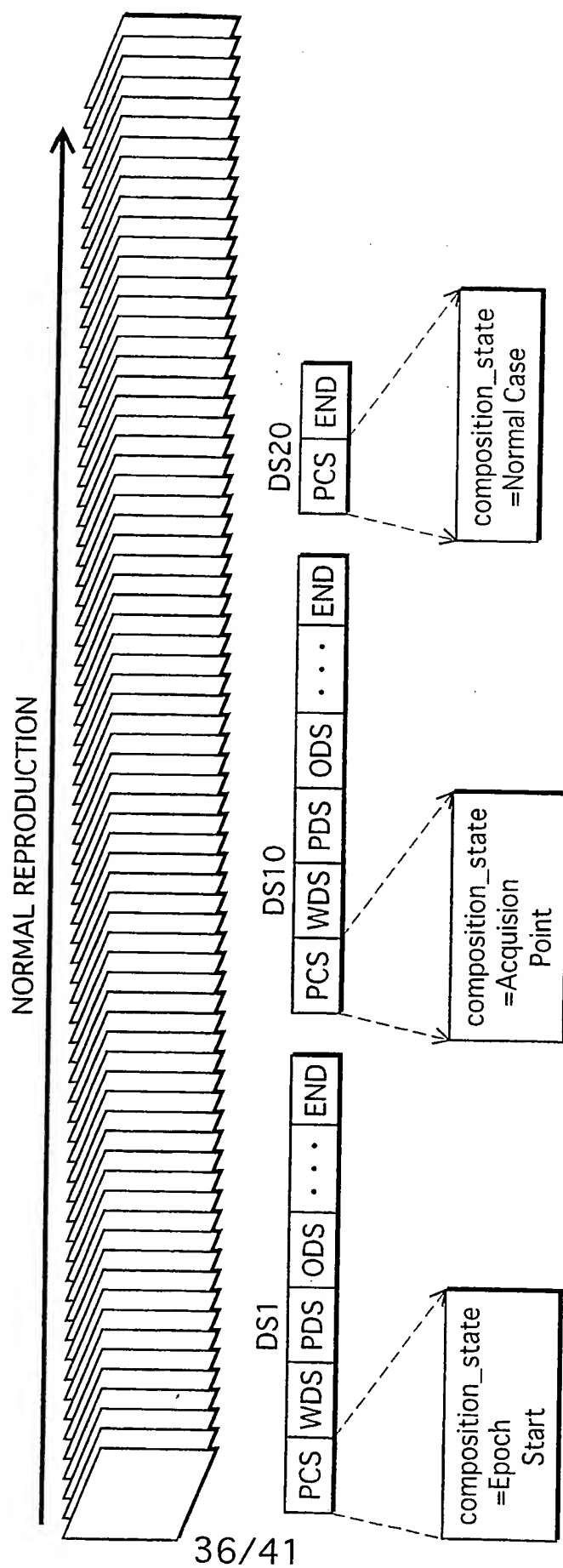


FIG.37

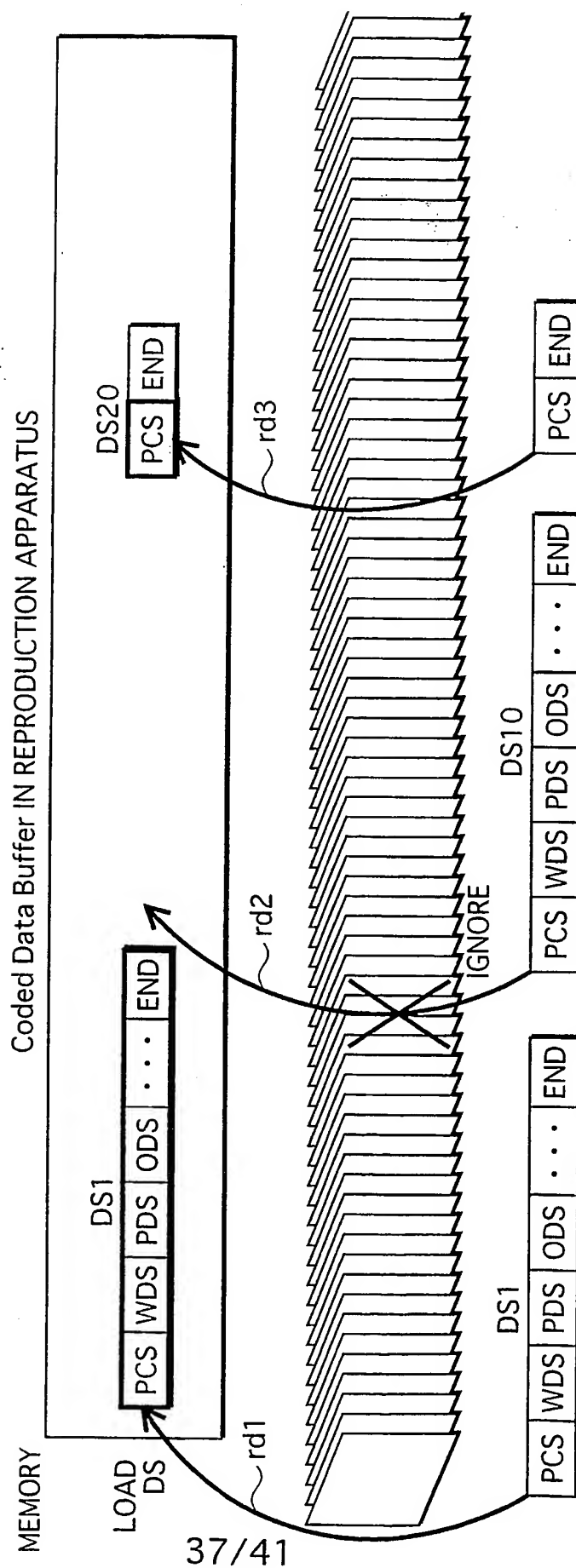


FIG.38

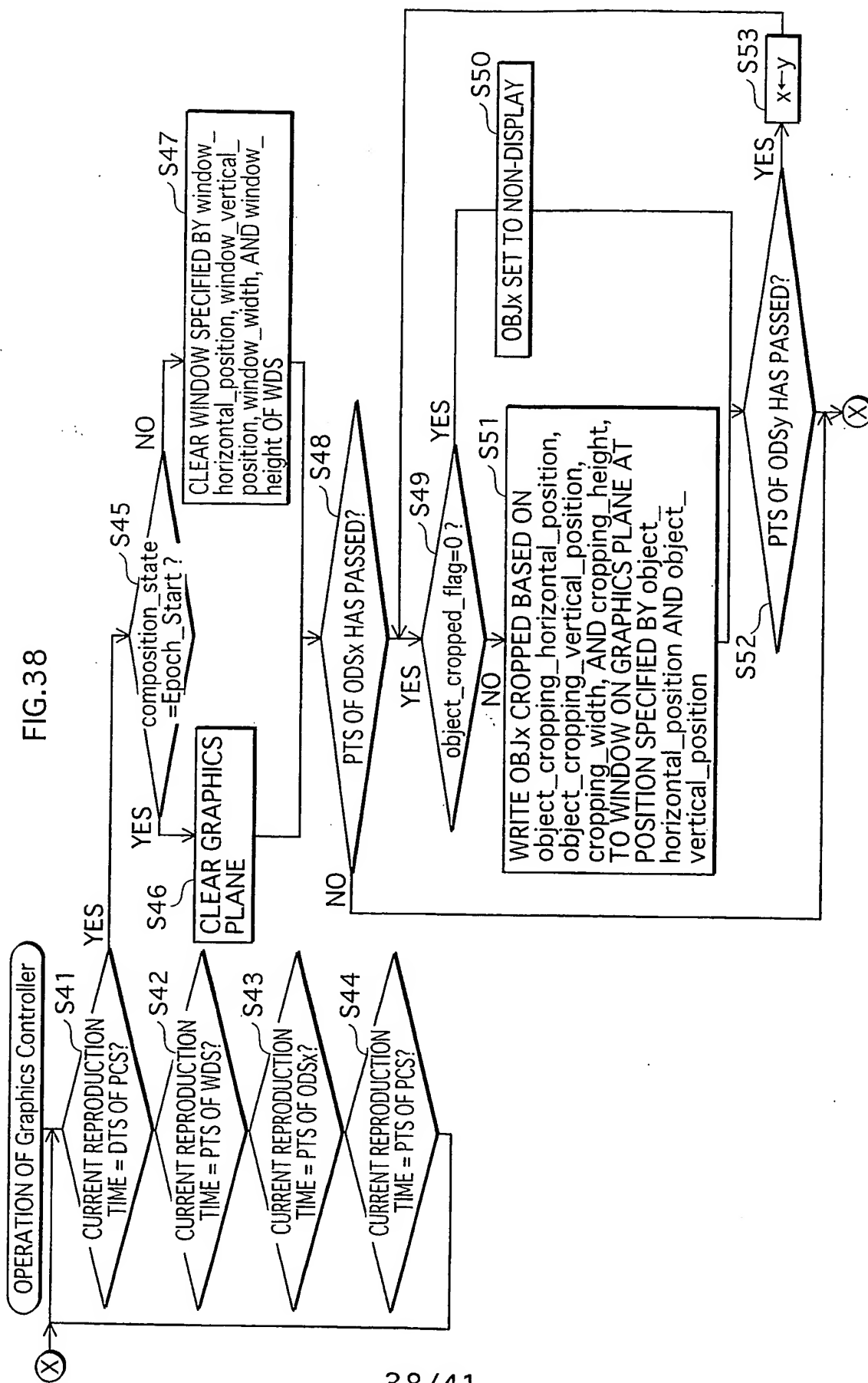


FIG.39

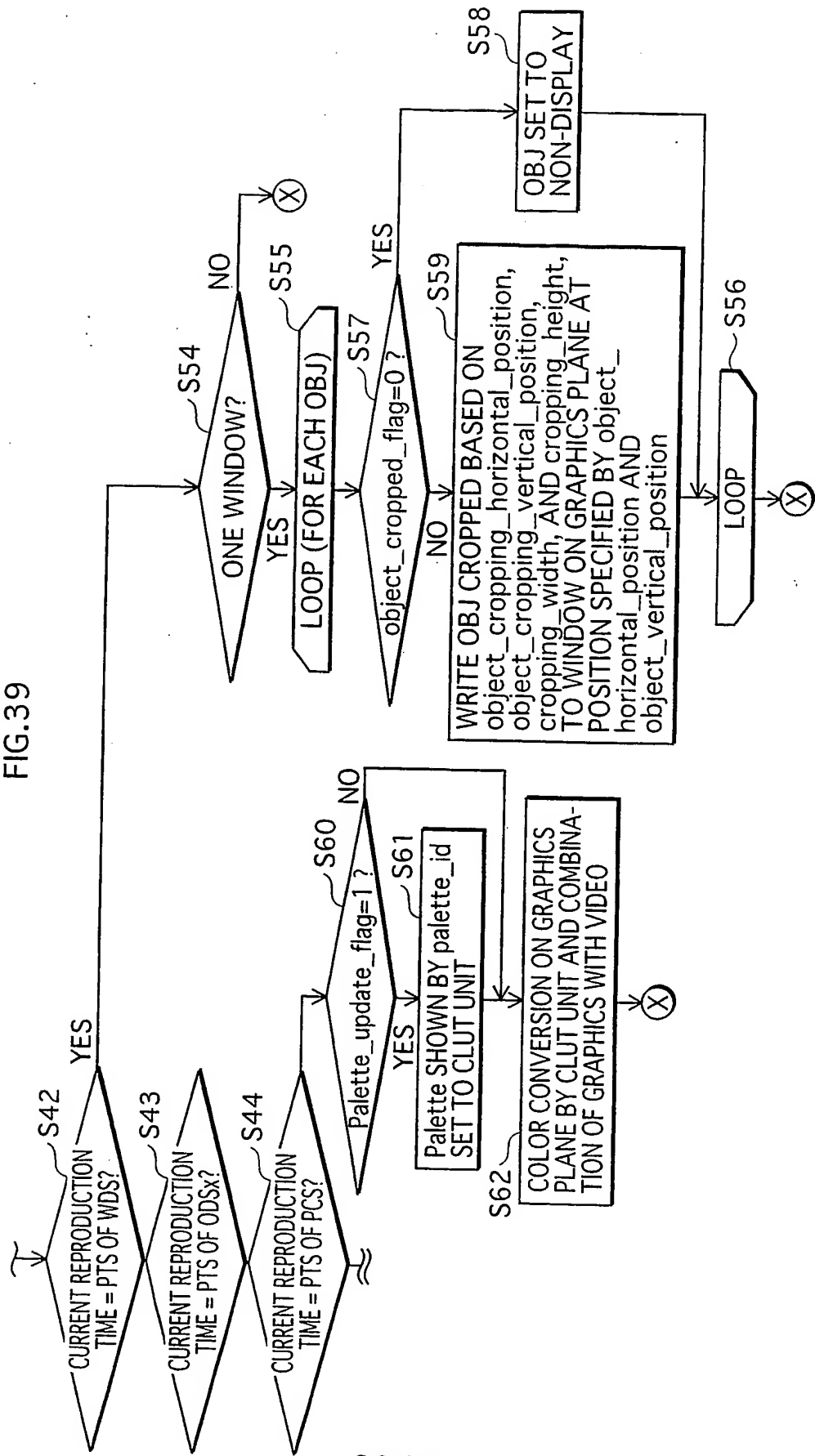


FIG.40

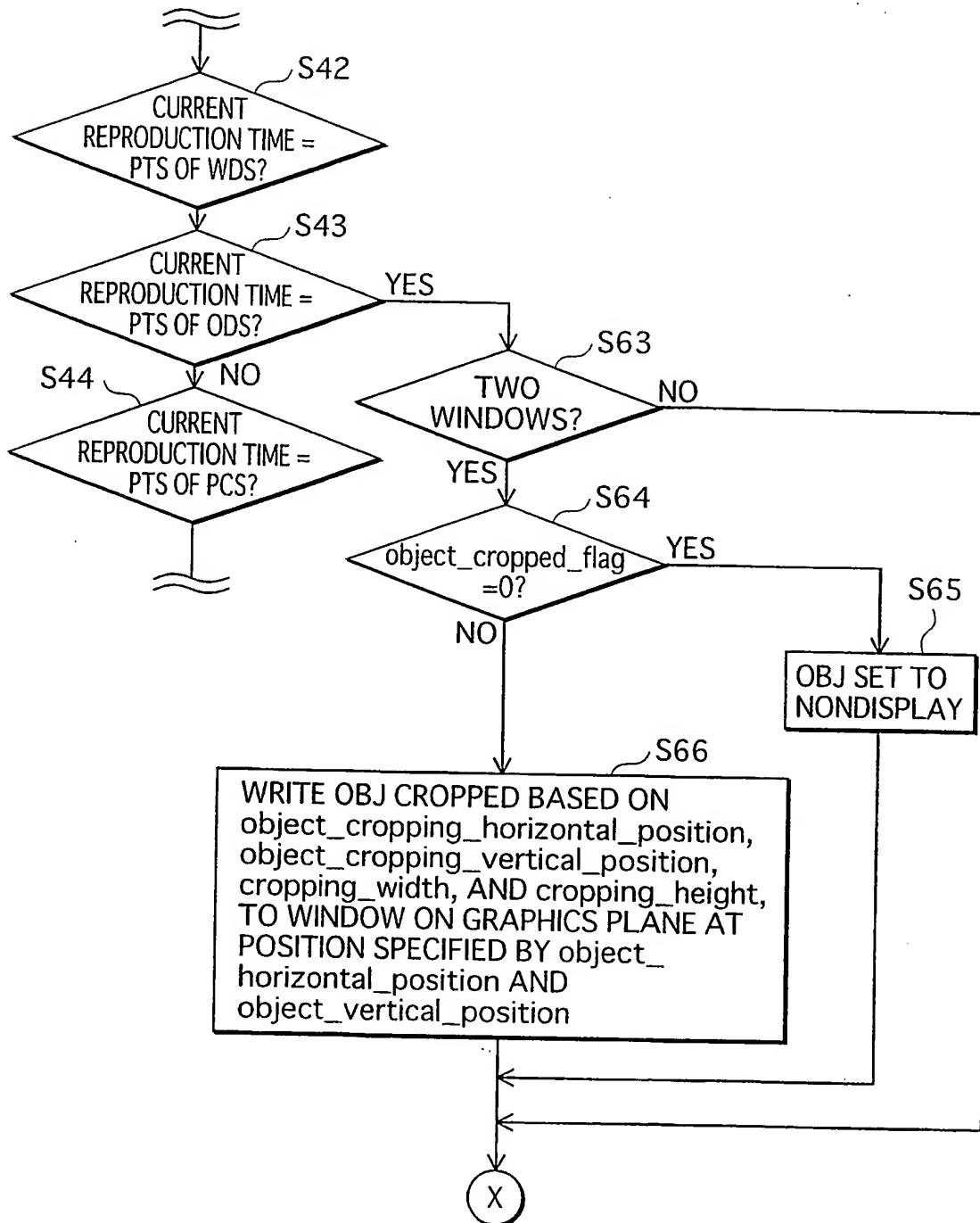
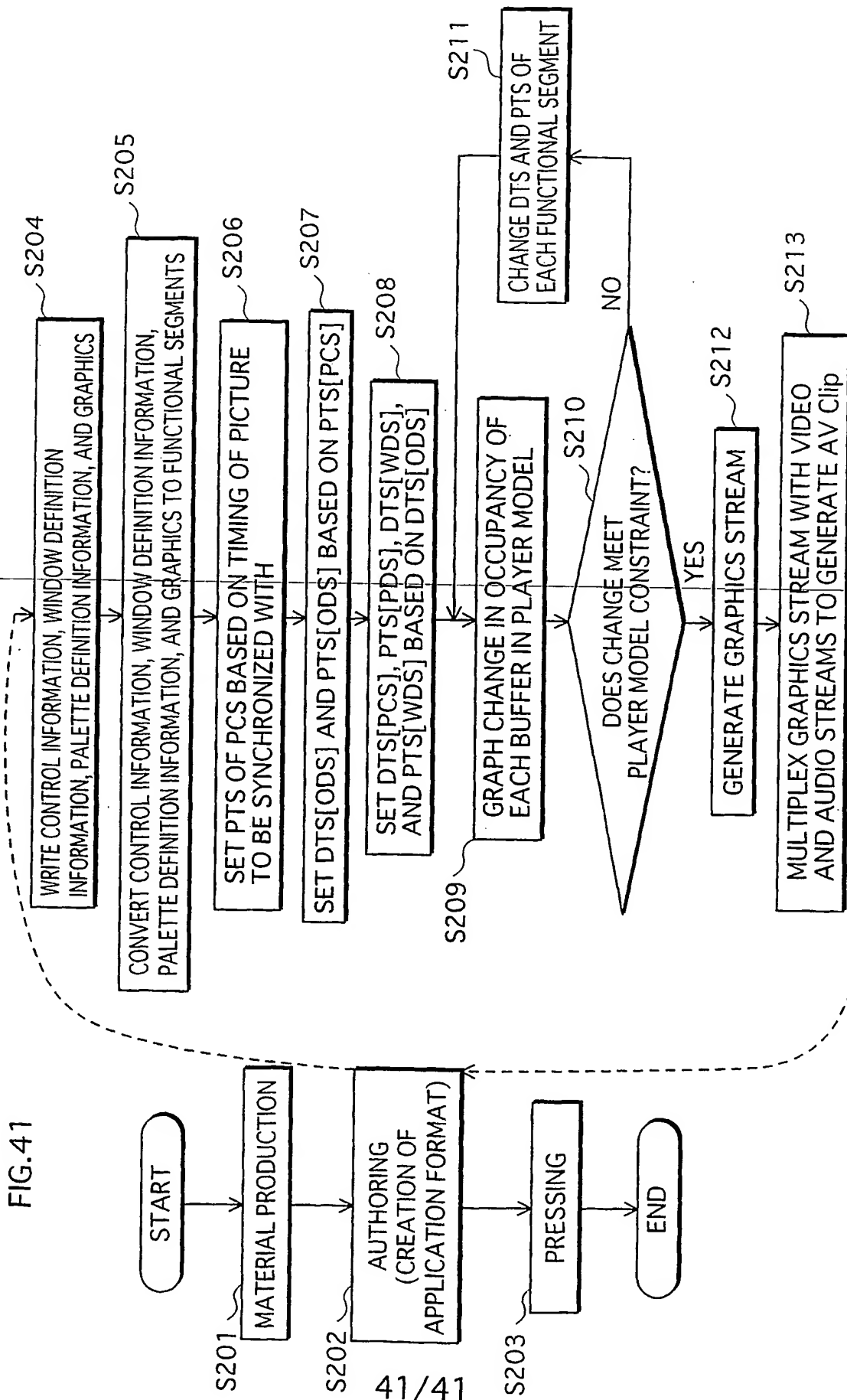


FIG. 41



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.